

วิธีการฮิวริสติกสำหรับปัญหาการรับและส่งพัสดุภายในวันทำการเดียวกัน



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมอุตสาหกรรม ภาควิชาวิศวกรรมอุตสาหกรรม
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A HEURISTIC APPROACH FOR THE SAME-DAY PICKUP AND DELIVERY PROBLEM



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Industrial Engineering

Department of Industrial Engineering

FACULTY OF ENGINEERING

Chulalongkorn University

Academic Year 2019

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	วิธีการฮิวริสติกสำหรับปัญหาการรับและส่งพัสดุภายในวันทำการเดียวกัน
โดย	นายวิษณุ ศรีตระกูล
สาขาวิชา	วิศวกรรมอุตสาหการ
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.พิศิษฐ์ จารุมนีโรจน์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัยเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

.....	คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)	
คณะกรรมการสอบวิทยานิพนธ์	
.....	ประธานกรรมการ
(รองศาสตราจารย์ ดร.วิภาวี ธรรมภรณ์พิลาศ)	
.....	อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.พิศิษฐ์ จารุมนีโรจน์)	
.....	กรรมการ
(รองศาสตราจารย์ ดร.นระเกณท์ พุ่มชูศรี)	
.....	กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.ชูเวช ชาญสง่าเวช)	

วิษญะ ศรีตระกูล : วิธีการฮิวริสติกสำหรับปัญหาการรับและส่งพัสดุภายในวันทำการเดียวกัน. (A HEURISTIC APPROACH FOR THE SAME-DAY PICKUP AND DELIVERY PROBLEM) อ.ที่ปรึกษาหลัก : ผศ. ดร.พิศิษฐ์ จารุมณีโรจน์

งานวิจัยนี้มุ่งเน้นการพัฒนาแบบจำลองทางคณิตศาสตร์ และวิธีการฮิวริสติกสำหรับปัญหาการรับและส่งพัสดุภายในวันทำการเดียวกัน (Deterministic SDPD) โดยปัญหาดังกล่าวจัดเป็นปัญหาการจัดเส้นทางเดินรถรูปแบบหนึ่ง ซึ่งรถสามารถเดินทางออกจากท่ารถได้ตามรอบรถ (wave) ที่กำหนดเท่านั้น นอกจากนี้เวลาการปรากฏ (arrival time) ของคำสั่งของลูกค้าทั้งหมดเป็นข้อมูลที่ทราบล่วงหน้าตั้งแต่ต้นวันทำการ ทั้งนี้คำสั่งของลูกค้าหนึ่งๆ จะประกอบไปด้วยจุดรับและจุดส่งพัสดุที่แตกต่างกัน และไม่จำเป็นต้องถูกมอบหมายให้อยู่ในรอบรถเดียวกัน แต่จำเป็นต้องจัดส่งพัสดุให้ทันภายในวันทำการ หากเลือกที่จะรับพัสดุของคำสั่งนั้นๆ ผู้วิจัยได้ทำการศึกษาผลกระทบของการเพิ่มจำนวนรอบรถในวันทำการที่มีต่อประสิทธิภาพของการหาคำตอบด้วยวิธีแม่นยำ (exact method) บนกลุ่มตัวอย่างที่สุ่มสร้างขึ้นจำนวน 40 ตัวอย่าง โดยผู้วิจัยพบว่า การเพิ่มขึ้นของจำนวนรอบรถในวันทำการส่งผลให้ประสิทธิภาพการหาคำตอบด้วยวิธีแม่นยำลดลงอย่างมาก ผู้วิจัยจึงได้ทำการพัฒนาฮิวริสติก Adaptive Large Neighborhood Search (ALNS) สำหรับปัญหาดังกล่าว ซึ่งเมื่อเปรียบเทียบกับวิธีการหาคำตอบแบบแม่นยำแล้ว ผู้วิจัยพบว่าคำตอบที่ได้จาก ALNS ส่วนใหญ่มีคุณภาพเทียบเท่า หรือดีกว่าคำตอบที่ได้จากวิธีแม่นยำภายใต้กรอบเวลาในการหาคำนวณที่จำกัด แต่กลับใช้เวลาในการหาคำตอบต่ำกว่ามาก

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สาขาวิชา วิศวกรรมอุตสาหการ
ปีการศึกษา 2562

ลายมือชื่อนิสิต
ลายมือชื่อ อ.ที่ปรึกษาหลัก

6070311321 : MAJOR INDUSTRIAL ENGINEERING

KEYWORD: ADAPTIVE LARGE NEIGHBORHOOD SEARCH, DYNAMIC DISPATCH
WAVES PROBLEM, SAMEDAY PICKUP AND DELIVERY PROBLEM

Wichaya Sritrakool : A HEURISTIC APPROACH FOR THE SAME-DAY PICKUP
AND DELIVERY PROBLEM. Advisor: Asst. Prof. Pisit Jarumaneeroj, Ph.D.

This research focuses on the development of an Integer Programming (IP) model and a Heuristic approach for the Deterministic Same-day Pickup and Delivery Problem (Deterministic SDPD), where a vehicle is allowed to leave the depot at specific time periods, called waves, and order arrivals are known in advance. Moreover, each customer order comprises of a pair of services - namely pickup and delivery - not necessarily be completed in the same wave but has to be delivered by the end of the day if the request has been accepted and picked. We explored how numbers of daily dispatch waves affected the IP optimization performance on 40 randomly generated instances. We found that, by increasing the numbers of daily dispatch waves, the IP optimization performance dropped tremendously as expected. Therefore, an Adaptive Large Neighborhood Search (ALNS) heuristic is proposed. When compared to the results from the IP solver, most ALNS solutions are equal or better in terms of solution quality; but, with tractable computational time.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Field of Study: Industrial Engineering

Student's Signature

Academic Year: 2019

Advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยความกรุณาของ ผู้ช่วยศาสตราจารย์ ดร.พิศิษฐ์ จารุมณีโรจน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้มอบโอกาส ความรู้ แนวความคิด และข้อเสนอแนะอันเป็นแนวทางเพื่อการปรับปรุงแก้ไข และพัฒนาวิทยานิพนธ์จนเสร็จสิ้นสมบูรณ์

สุดท้ายนี้ ขอกราบขอบพระคุณ บิดา มารดา ผู้ให้กำเนิด และบุคคลรอบข้าง ที่คอยให้กำลังใจ และเป็นแรงกระตุ้นให้ผู้วิจัยทำวิทยานิพนธ์ฉบับนี้จนสำเร็จลุล่วงไปด้วยดี

วิษณุ ศรีตระกูล



สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ค
บทคัดย่อภาษาอังกฤษ.....	ง
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉ
สารบัญตาราง.....	ฌ
สารบัญรูปภาพ.....	ญ
บทที่ 1 บทนำ.....	1
1.1 ภาพรวมและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	4
1.3 ขอบเขตของงานวิจัย.....	5
1.4 ผลที่ได้รับจากงานวิจัย.....	5
บทที่ 2 การทบทวนวรรณกรรม.....	6
2.1 ปัญหาการเดินทางของพนักงานขาย และปัญหาการจัดเส้นทางเดินรถ (TSP and VRP).....	6
2.1.1 การสร้างคำตอบเบื้องต้น (Constructive heuristics).....	9
2.1.2 การปรับปรุงคำตอบ (Improvement heuristics).....	9
2.2 Dynamic Dispatch Waves Problem (DDWP).....	11
2.2.1 การถ่วงดุลของการปล่อยรถของปัญหา DDWP.....	12
2.2.2 การถ่วงดุลของจำนวนลูกค้าที่ให้บริการลูกค้าต่อเที่ยวรถของปัญหา DDWP.....	12
2.3 Pickup and Delivery Problem with Cross-Docking (PDPCD).....	13
2.4 Adaptive Large Neighborhood Search (ALNS).....	13
2.5 งานวิจัยที่เกี่ยวข้อง.....	14

2.5.1	The Dynamic Dispatch Waves Problem for same-day delivery	14
2.5.2	The Pickup and Delivery Problem with Cross-Docking	15
2.5.3	Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests	16
บทที่ 3	แนวทางการแก้ปัญหา	17
3.1	ลักษณะของปัญหา.....	17
3.2	สมมติฐานของปัญหา	18
3.3	แนวทางการดำเนินงาน	19
3.3.1	การตั้งค่าของปัญหา SDPD.....	19
3.3.2	การสร้างแบบจำลองทางคณิตศาสตร์สำหรับปัญหา SDPD แบบ Deterministic	21
3.3.3	การพัฒนาฮิวริสติกสำหรับปัญหา SDPD	28
3.3.5	ตัวอย่างการหาคำตอบของฮิวริสติก	46
บทที่ 4	ผลการดำเนินงานวิจัย	50
4.1	ข้อมูลนำเข้าที่ใช้ในการหาคำตอบ	50
4.2	การทดลองเพื่อหาขนาดปัญหาที่เหมาะสมกับสมมติฐาน.....	51
4.3	การทดลองเพื่อหาขีดจำกัดความสามารถของวิธีแมนตรง.....	54
4.4	การตั้งค่าของฮิวริสติก.....	55
4.5	ผลการเปรียบเทียบคุณภาพของคำตอบ และเวลาในการหาของคำตอบของฮิวริสติก	56
4.5.1	ประสิทธิภาพของฮิวริสติกในการหาคำตอบของปัญหาที่มี 6 รอบรถ	57
4.5.2	ประสิทธิภาพของฮิวริสติกในการหาคำตอบของปัญหาที่มี 8 รอบรถ	58
4.5.3	ประสิทธิภาพของฮิวริสติกในการหาคำตอบของปัญหาที่มี 10 รอบรถ	59
4.5.4	ประสิทธิภาพของฮิวริสติกในการหาคำตอบของปัญหาที่มี 12 รอบรถ	61
บทที่ 5	สรุปผลการดำเนินงานวิจัย	63
5.1	ผลการวิจัย.....	63

5.2 แนวทางสำหรับพัฒนาวิธีการแก้ปัญหา SDPD ที่มีความไม่แน่นอนของคำสั่งของลูกค้า.....	64
5.3 ข้อเสนอแนะ	65
บรรณานุกรม.....	67
ประวัติผู้เขียน.....	71



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญตาราง

	หน้า
ตารางที่ 2-1 รายละเอียดของเส้นทางเดินรถในตัวอย่างคำตอบสำหรับปัญหา SDPD.....	8
ตารางที่ 3-1 เซต และพารามิเตอร์ทั่วไปในแบบจำลองทางคณิตศาสตร์.....	22
ตารางที่ 3-2 เซต และพารามิเตอร์ของรอบรถในแบบจำลองทางคณิตศาสตร์.....	22
ตารางที่ 3-3 ตัวอย่างข้อมูลนำเข้าของปัญหา SDPD.....	29
ตารางที่ 3-4 พารามิเตอร์สำหรับให้คะแนน Operator ใน ALNS.....	45
ตารางที่ 3-5 ข้อมูลนำเข้าของตัวอย่างการหาคำตอบด้วยฮิวริสติก.....	46
ตารางที่ 3-6 การตั้งค่าของ ALNS ในตัวอย่างการหาคำตอบด้วยฮิวริสติก.....	47
ตารางที่ 4-1 เวลาในการหาคำตอบในหน่วยวินาทีของปัญหาขนาดต่างกันที่มี 6 รอบรถ (วินาที)....	53
ตารางที่ 4-2 %GAP ของการหาคำตอบของปัญหาขนาดต่างกันที่มี 6 รอบรถ.....	53
ตารางที่ 4-3 เวลาในการหาคำตอบของปัญหาขนาด 16 คำสั่งด้วยวิธีแมนตรง (วินาที).....	54
ตารางที่ 4-4 %GAP ของการหาคำตอบของปัญหาขนาด 16 คำสั่งที่ด้วยวิธีแมนตรง.....	55
ตารางที่ 4-5 การตั้งค่าของฮิวริสติก.....	56
ตารางที่ 4-6 ผลการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 6 รอบรถ.....	57
ตารางที่ 4-7 เวลาในการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 6 รอบรถ (วินาที).....	58
ตารางที่ 4-8 ผลการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 8 รอบรถ.....	58
ตารางที่ 4-9 เวลาในการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 8 รอบรถ (วินาที).....	59
ตารางที่ 4-10 ผลการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 10 รอบรถ.....	60
ตารางที่ 4-11 เวลาในการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 10 รอบรถ (วินาที).....	60
ตารางที่ 4-12 ผลการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 12 รอบรถ.....	61
ตารางที่ 4-13 เวลาในการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 12 รอบรถ (วินาที).....	62

สารบัญรูปร่างภาพ

หน้า

รูปที่ 1-1 มูลค่าการขายปลีก e-commerce ของโลกตั้งแต่ปีค.ศ. 2014 ถึงปีค.ศ. 2021 (Statista).	1
รูปที่ 1-2 นโยบายการปล่อยรถของการบริการรับและส่งพัสดุในวันทำการเดียวกัน	3
รูปที่ 1-3 รูปแบบปัญหาการจัดเส้นทางรถ	3
รูปที่ 2-1 ปัญหาการเดินทางของพนักงานขาย และปัญหาการจัดเส้นทางเดินรถ.....	6
รูปที่ 2-2 ตัวอย่างคำตอบของปัญหา SDPD	7
รูปที่ 2-3 แผนการเดินทางของตัวอย่างคำตอบสำหรับปัญหา SDPD	8
รูปที่ 2-4 การปรับปรุงคำตอบภายในเส้นทางเดียวกัน (Intra-route Improvements)	10
รูปที่ 2-5 การปรับปรุงคำตอบจากคนละเส้นทาง (Inter-route Improvements).....	11
รูปที่ 2-6 ตัวอย่างการปล่อยรถในปัญหา DDWP.....	12
รูปที่ 3-1 ช่วงเวลาทำการของปัญหา SDPD	19
รูปที่ 3-2 ข้อกำหนดในการจัดเส้นทางของปัญหา SDPD	20
รูปที่ 3-3 ตัวอย่างเวลาการปรากฏ และรอบการปรากฏของคำสั่ง	20
รูปที่ 3-4 ตัวอย่างการทำงานของอสมการข้อจำกัดที่ i.....	25
รูปที่ 3-5 ตัวอย่างการทำงานของอสมการข้อจำกัดที่ ii.....	26
รูปที่ 3-6 ตัวอย่างการทำงานของอสมการข้อจำกัดที่ vi. ถึง viii.	27
รูปที่ 3-7 ตัวอย่างการทำงานของอสมการข้อจำกัดที่ ix. ถึง x.....	27
รูปที่ 3-8 แผนภาพแสดงขั้นตอนการทำงานของ ALNS.....	30
รูปที่ 3-9 Pseudocode แสดงขั้นตอนการทำงานของ ALNS	31
รูปที่ 3-10 แผนภาพแสดงขั้นตอนการสร้างคำตอบเริ่มต้น	33
รูปที่ 3-11 Pseudocode แสดงขั้นตอนการสร้างคำตอบเริ่มต้น	33
รูปที่ 3-12 แผนภาพแสดงขั้นตอนการทำลายคำตอบ	34

รูปที่ 3-13 ตัวอย่างการทำลายคำตอบ	35
รูปที่ 3-14 Pseudocode แสดงขั้นตอนการทำงานของ Random removal.....	36
รูปที่ 3-15 Pseudocode แสดงขั้นตอนการทำงานของ Least prize removal	36
รูปที่ 3-16 Pseudocode แสดงขั้นตอนการทำงานของ Shaw removal.....	37
รูปที่ 3-17 แผนภาพแสดงขั้นตอนการซ่อมแซมคำตอบ	37
รูปที่ 3-18 ตัวอย่างการซ่อมแซมคำตอบ	39
รูปที่ 3-19 Pseudocode แสดงขั้นตอนการทำงานของ Greedy insertion.....	40
รูปที่ 3-20 Pseudocode แสดงขั้นตอนการทำงานของ Regret-2 insertion.....	41
รูปที่ 3-21 Pseudocode แสดงขั้นตอนการทำงานของ Regret-3 insertion.....	42
รูปที่ 3-22 แผนภาพแสดงขั้นตอนการทำงานของ Local search	43
รูปที่ 3-23 Pseudocode แสดงขั้นตอนการทำงานของ Local search.....	43
รูปที่ 3-24 Pseudocode แสดงขั้นตอนการทำงานของ Relocation	44
รูปที่ 3-30 ตัวอย่างคำตอบที่ได้จากฮิวริสติก	48
รูปที่ 3-31 ค่า Objective ของคำตอบที่ถูกยอมรับโดย SA.....	48
รูปที่ 3-32 อุณหภูมิของ SA ของรอบการหาคำตอบที่มีการยอมรับด้วย SA	49
รูปที่ 4-1 ขอบเขตการให้บริการ และค่าส่งกรณีที่แย่ที่สุด.....	51
รูปที่ 4-2 ตัวอย่างคำตอบของปัญหา SDPD ที่ระดับการให้บริการถูกจำกัดด้วยระยะเวลาทำการ ...	51
รูปที่ 4-3 ระดับการให้บริการเฉลี่ย และ %GAP เฉลี่ยของคำตอบจากปัญหา SDPD ที่มี 6 รอบรถ	52
รูปที่ 5-1 ตัวอย่างคำตอบของปัญหา Stochastic SDPD	65

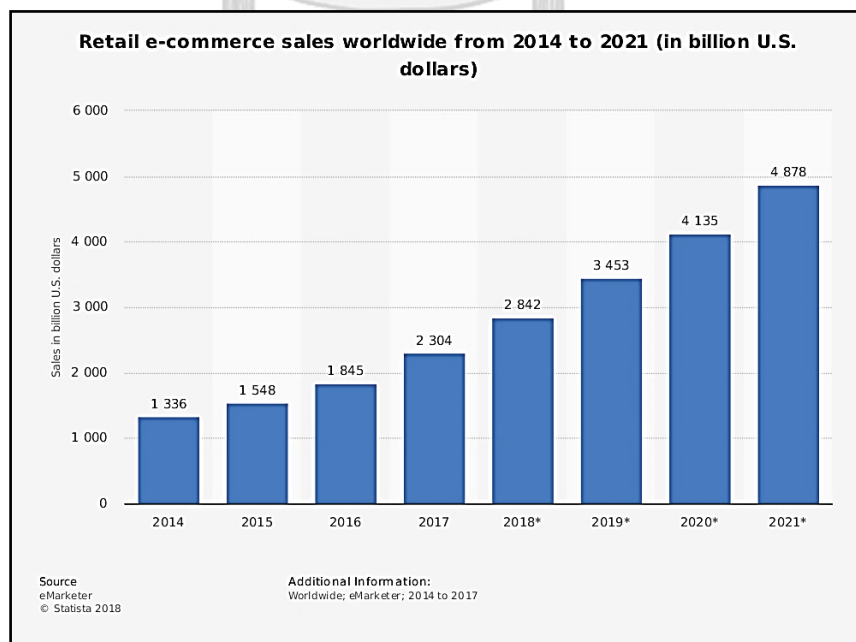
บทที่ 1

บทนำ

ในบทนี้จะกล่าวถึงภาพรวม และความสำคัญของปัญหาการรับและส่งพัสดุภายในวันทำการเดียวกัน รวมไปถึงแนวทางการสร้างแบบจำลองทางคณิตศาสตร์ และการพัฒนาวิธีการฮิวริสติกสำหรับปัญหาดังกล่าว โดยพิจารณาถึงองค์ประกอบต่าง ๆ ของปัญหาอย่างเป็นลำดับขั้น

1.1 ภาพรวมและความสำคัญของปัญหา

จากการสำรวจ และคาดการณ์มูลค่าการค้าปลีกของ e-commerce ตั้งแต่ปีค.ศ. 2014 จนถึงปีค.ศ. 2021 ดังแสดงในรูปที่ 1-1 แสดงให้เห็นถึงแนวโน้มการค้าผ่านช่องทางออนไลน์ที่เพิ่มสูงขึ้นอย่างต่อเนื่อง ส่งผลทำให้ผู้ประกอบการจำเป็นต้องขนส่งสินค้าไปยังลูกค้าปลายทางมากขึ้น บริษัทตัวแทนขนส่งเป็นหนึ่งในทางเลือกที่ผู้ประกอบการนิยมใช้ เนื่องจากเป็นทางเลือกที่มีต้นทุนคงที่ (Fixed cost) ต่ำกว่าการขนส่งโดยบริษัทของผู้ประกอบการเอง บริษัทไม่จำเป็นต้องลงทุนค่าพาหนะขนส่ง และค่าจ้างพนักงานขับรถ ซึ่งมีความเหมาะสมกับ การส่งสินค้าของร้านค้าปลีกออนไลน์ที่ไม่มีระบบขนส่งเป็นของตัวเอง



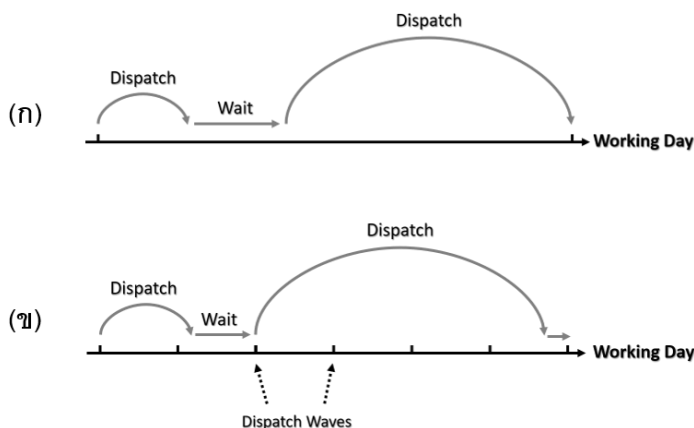
รูปที่ 1-1 มูลค่าการค้าขายปลีก e-commerce ของโลกตั้งแต่ปีค.ศ. 2014 ถึงปีค.ศ. 2021 (Statista)

บริการรับและส่งพัสดุ (Pickup and Delivery Service) เป็นบริการรูปแบบหนึ่งของบริษัท ตัวแทนขนส่ง ที่ได้รับความนิยมจากลูกค้าที่ไม่ประสงค์เดินทางไปส่งพัสดุที่สาขาของบริษัทด้วยตนเอง ซึ่งบริการลักษณะนี้เริ่มเป็นที่นิยมมากขึ้นในประเทศไทย เนื่องจากเป็นการบริการที่สะดวก ช่วยประหยัดทั้งเวลา และค่าใช้จ่ายในการเดินทางไปยังสาขาของบริษัทขนส่ง

สำหรับบริการรับและส่งพัสดุโดยทั่วไปนั้น พนักงานขนส่งจะเดินทางจากท่ารถไปรับพัสดุ ภายในวันที่ลูกค้าทำการจองไว้ แล้วจึงนำพัสดุกลับมายังคลังพัสดุ เพื่อเตรียมจัดส่งในวันทำการถัดไป (Next-day Pickup and Delivery) อย่างไรก็ตาม บริษัทตัวแทนขนส่งบางแห่งได้เริ่มเปิดให้บริการรับและส่งพัสดุภายในวันทำการเดียวกัน (Same-day Pickup and Delivery: SDPD) เพื่อตอบสนองต่อความต้องการของลูกค้าในสังคมปัจจุบันที่มีความรีบเร่ง โดยพนักงานขนส่งจะเดินทางจากท่ารถไปรับพัสดุภายในช่วงเช้าของวันที่ลูกค้านัดจองไว้ แล้วจึงจัดส่งพัสดุไปยังจุดหมายปลายทางภายในวันทำการเดียวกัน ตัวอย่างการให้บริการรูปแบบนี้ เช่น บริการ Bangkok same day ของบริษัท Kerry Express เป็นต้น

บริษัท Kerry Express เปิดให้บริการ Bangkok same day เมื่อเดือนมิถุนายนในปี.ศ.2558 โดยบริการ Bangkok same day ถือเป็นหนึ่งทางเลือกสำหรับการขนส่งพัสดุแบบเร่งด่วน ที่มุ่งเน้นการอำนวยความสะดวกแก่กลุ่มลูกค้าภายในจังหวัดกรุงเทพมหานคร เมื่อลูกค้าจองรถผ่านบริการสายด่วน หรือแอปพลิเคชันของทางบริษัทเป็นที่เรียบร้อยแล้ว เจ้าหน้าที่จะเดินทางไปรับพัสดุจากลูกค้า ภายในช่วงเช้าของวันทำการ แล้วเดินทางกลับมายังสาขาหลักที่มีหน้าที่เป็นท่าเปลี่ยนถ่ายสินค้า (Cross dock) ก่อนเวลาบ่ายโมงตรง เพื่อรวบรวม และเตรียมจัดส่งพัสดุไปยังจุดหมายปลายทางที่ลูกค้ากำหนดภายในระยะเวลาทำการของบริษัท

การบริการรับและส่งพัสดุในวันทำการเดียวกันจะมีการปรากฏขึ้นของคำสั่งของลูกค้าอย่างต่อเนื่อง ดังนั้นการใช้นโยบายปล่อยรถแบบมีรอบรถ (Wave-based dispatching) จึงสามารถตอบสนองต่อคำสั่งของลูกค้าได้อย่างเป็นระบบกว่าการปล่อยรถแบบไม่มีรอบรถ (On-line dispatching) ดังแสดงในรูปที่ 1-2(ข) ซึ่งวันทำการถูกแบ่งออกเป็นรอบรถจำนวน 6 รอบ โดยในแต่ละรอบรถจะมีการเลือก และกำหนดเส้นทางคำสั่งของลูกค้าที่สามารถให้บริการในรอบรถนั้น ๆ เพื่อทยอยดำเนินการตามคำสั่งของลูกค้าที่ปรากฏขึ้น

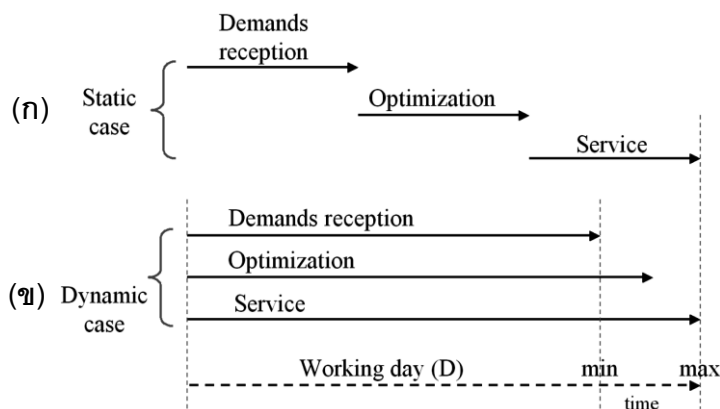


รูปที่ 1-2 นโยบายการปล่อยรถของการบริการรับและส่งพัสดุในวันทำการเดียวกัน

(ก) การปล่อยรถแบบไม่มีรอปรรถ และ (ข) การปล่อยรถแบบมีรอปรรถ

นอกจากนี้ การปล่อยรถอย่างไม่มีประสิทธิภาพอาจก่อให้เกิดค่าใช้จ่ายที่สูงเกินควร ตัวอย่างเช่น การปล่อยรถด้วยความถี่สูงเกินไปอาจทำให้เกิดค่าใช้จ่ายในการเดินทางสูง เนื่องจากการปล่อยรถแต่ละครั้งไม่คุ้มที่ควร อย่างไรก็ตาม การปล่อยรถด้วยความถี่ต่ำเกินไป อาจทำให้ไม่สามารถบริการลูกค้าบางส่วนได้ทันเวลา เนื่องจากการสะสมของคำสั่งของลูกค้าในระบบมาก อันเป็นผลมาจากการขาดรถเป็นเวลานาน

โดยทั่วไปแล้ว ปัญหาการรับและส่งพัสดุในวันทำการเดียวกัน (Same-day Pickup and Delivery Problem: SDPD) เป็นปัญหาการจัดเส้นทางรถแบบพลวัต (Dynamic Vehicle Routing Problem: DVRP) โดยมีการรับ และส่งพัสดุ ที่มีความไม่แน่นอน เนื่องจากต้องรอรับคำสั่ง (Demands reception) แล้วจึงดำเนินการจัดเส้นทางที่ดีที่สุดสำหรับการปล่อยรถในแต่ละรอบ (Optimization and Service) อย่างต่อเนื่องตลอดวันทำการ ดังแสดงในรูปที่ 1-3(ข)



รูปที่ 1-3 รูปแบบปัญหาการจัดเส้นทางรถ

(ก) SVRP และ (ข) DVRP (Hajjam et al., 2013)

การพัฒนาวิธีการหาคำตอบสำหรับปัญหา SDPD มักเริ่มจากการศึกษาปัญหาในรูปแบบ Deterministic ที่ทราบเวลาการปรากฏของคำสั่งทั้งหมดภายในวันทำการก่อน จากนั้นจึงพัฒนาวิธีการแก้ปัญหาในรูปแบบ Stochastic ที่มีความไม่แน่นอนของคำสั่งของลูกค้า หรือเพิ่มข้อจำกัดด้านทรัพยากรอื่น ๆ ร่วมด้วย ยกตัวอย่างเช่น งานวิจัยเรื่อง The Dynamic Dispatch Waves Problem for same-day delivery ของ Klapp et al., (2018) ซึ่งทำการศึกษาปัญหาการส่งพัสดุในวันทำการเดียวกัน (Same-day Delivery: SDD) และได้พัฒนาต้นแบบ Arc-based Integer Programming สำหรับปัญหา SDD โดยอ้างอิงจากปัญหา Dynamic Dispatch Waves Problem (DDWP) แบบ Deterministic แล้วนำคำตอบที่ได้จากการแก้แบบจำลองทางคณิตศาสตร์ไปพัฒนาวิธีการหาคำตอบเบื้องต้น และนโยบายฮิวริสติกที่สามารถลดค่าใช้จ่ายเฉลี่ยจากการดำเนินงานของปัญหา SDD ที่มีความไม่แน่นอนของคำสั่งของลูกค้าในลำดับถัดไป

เนื่องจากปัญหา SDD ที่ Klapp et al., (2018) ได้ทำการศึกษานั้น มีลักษณะเดียวกับกับปัญหา SDPD ในมุมมองของการบริหารจัดการทรัพยากรเวลา และรถที่มีอยู่อย่างจำกัด เพื่อตอบสนองต่อคำสั่งของลูกค้าที่มีความไม่แน่นอน แต่มีข้อแตกต่างที่สำคัญ คือ คำสั่งของลูกค้าในปัญหา SDPD ประกอบด้วย 2 ส่วน ได้แก่ การรับพัสดุ และการส่งพัสดุ ในขณะที่คำสั่งของลูกค้าในปัญหา SDD มีเพียงการส่งพัสดุเท่านั้น ผู้วิจัยจึงได้ทำการพัฒนาแบบจำลองทางคณิตศาสตร์สำหรับปัญหา SDPD โดยอ้างอิงจากปัญหา DDWP และปัญหาการรับและส่งพัสดุ (Pickup and Delivery Problem: PDP) นอกจากนี้ ผู้วิจัยยังได้ทำการพัฒนาวิธีการฮิวริสติกสำหรับปัญหา SDPD แบบ Deterministic เพื่อใช้เป็นแนวทางเบื้องต้นสำหรับการพัฒนาวิธีการหาคำตอบของปัญหา SDPD ที่มีความไม่แน่นอนของคำสั่งของลูกค้า (Stochastic SDPD) หรือปัญหา SDPD ที่มีข้อจำกัดด้านทรัพยากรอื่น ๆ ซึ่งมีความใกล้เคียงกับปัญหาจริงมากยิ่งขึ้นในอนาคต ทั้งนี้ ผู้วิจัยได้เลือกใช้ฮิวริสติก Adaptive Large Neighborhood Search (ALNS) ในงานวิจัยฉบับนี้ เนื่องด้วยฮิวริสติกดังกล่าวเป็นที่นิยมอย่างแพร่หลายในการหาคำตอบของปัญหา PDP รูปแบบต่าง ๆ (Li et al., 2016, Ghilas et al., 2016, Sun et al., 2020)

1.2 วัตถุประสงค์ของงานวิจัย

สร้างแบบจำลองทางคณิตศาสตร์สำหรับปัญหา SDPD แบบ Deterministic ที่มีจำนวนรถจำกัด เพื่อเพิ่มกำไรสุทธิจากการดำเนินงาน และการพัฒนาวิธีการฮิวริสติกสำหรับปัญหา SDPD แบบ Deterministic ที่สามารถให้คำตอบที่มีคุณภาพ โดยใช้เวลาเฉลี่ยในการหาคำตอบน้อยกว่าวิธีการแม่นยำ (Exact method)

1.3 ขอบเขตของงานวิจัย

ขอบเขตของงานวิจัยประกอบไปด้วย

- ทราบเวลาการปรากฏของคำสั่งของลูกค้าทั้งหมดภายในวันทำการ
- จำนวนรถที่ใช้งานได้มีหนึ่งคัน และมีความจุของรถไม่จำกัด
- จำนวนท่ารถมีหนึ่งท่า
- ทุกเส้นทางการเดินรถต้องเริ่มต้น และสิ้นสุดที่ท่ารถเสมอ
- สามารถปล่อยรถออกจากท่ารถ ณ รอบรถใดรอบรถหนึ่งเท่านั้น
- รถต้องกลับมายังท่ารถก่อนหมดเวลาทำการเสมอ
- ทุกคำสั่งของลูกค้ามีค่าตอบแทน (Order prize) ซึ่งเป็นรายรับที่สมมติขึ้นเพื่อให้เกิดความจำเป็นในการบริการลูกค้าแต่ละราย

1.4 ผลที่ได้รับจากงานวิจัย

- แบบจำลองทางคณิตศาสตร์สำหรับปัญหา SDPD ที่ทราบเวลาการปรากฏของคำสั่งของลูกค้าทั้งหมดภายในวันทำการ เพื่อเพิ่มกำไรสุทธิจากการดำเนินงาน
- อัลกอริทึมสำหรับปัญหา SDPD ที่สามารถให้คำตอบที่มีคุณภาพ แต่ใช้เวลาเฉลี่ยในการหาคำตอบน้อยกว่าวิธีการแม่นยำตรง (Exact method)
- แนวทางสำหรับพัฒนาวิธีการแก้ปัญหา SDPD ที่มีความไม่แน่นอนของคำสั่งของลูกค้า

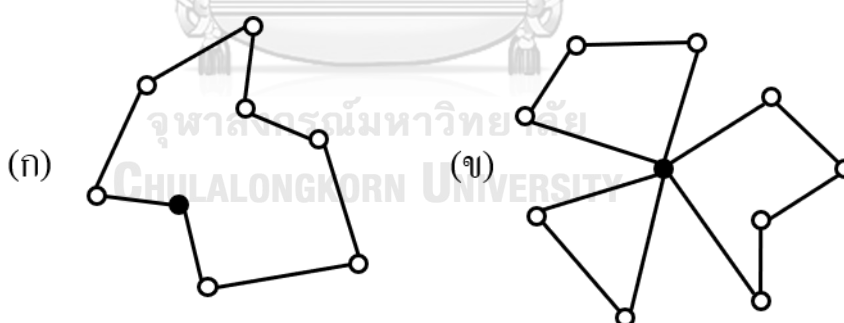
บทที่ 2

การทบทวนวรรณกรรม

ในบทนี้จะนำเสนอทฤษฎีที่เกี่ยวข้องกับการสร้างแบบจำลองทางคณิตศาสตร์ และการพัฒนาฮิวริสติก สำหรับปัญหา SDPD รวมถึงงานวิจัยที่เกี่ยวข้อง เพื่อใช้เป็นแนวทางในการดำเนินการวิจัย ซึ่งมีเนื้อหาเกี่ยวข้องกับ ปัญหาการเดินทางของพนักงานขาย และปัญหาการจัดเส้นทางเดินรถ (TSP and VRP), Dynamic Dispatch Waves Problem (DDWP), Vehicle Routing Problem with Crossdocking (VRPCD), Adaptive Large Neighborhood Search (ALNS), และงานวิจัยที่เกี่ยวข้อง

2.1 ปัญหาการเดินทางของพนักงานขาย และปัญหาการจัดเส้นทางเดินรถ (TSP and VRP)

ปัญหาการเดินทางของพนักงานขาย (Travelling Salesman Problem; TSP) มุ่งเน้นการจัดเส้นทางให้พนักงานขายเดินทางไปพบลูกค้าทุกราย และกลับมายังจุดเริ่มต้น โดยมีระยะทางในการเดินทาง หรือมีค่าใช้จ่ายจากการเดินทางน้อยที่สุด ดังแสดงในรูปที่ 2-1(ก) เมื่อจุดสีดำ และจุดสีขาว คือ ตำแหน่งของท่ารถ (Depot) และตำแหน่งของลูกค้า ตามลำดับ (Mahi et al., 2015, Cacchiani et al., 2020, Hougardy et al., 2020)



รูปที่ 2-1 ปัญหาการเดินทางของพนักงานขาย และปัญหาการจัดเส้นทางเดินรถ

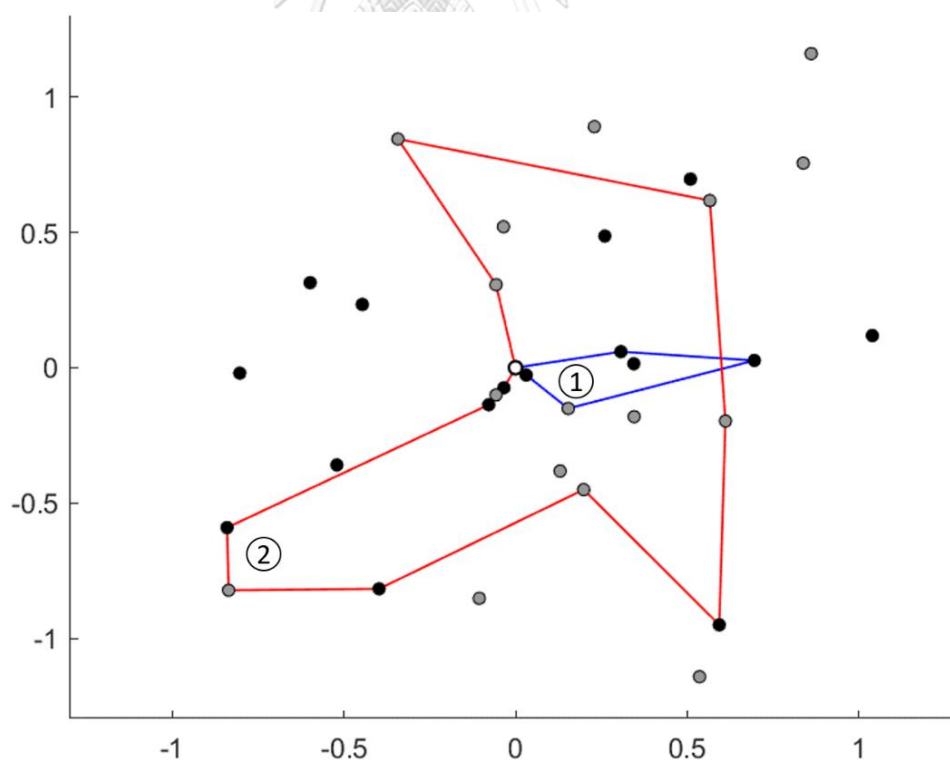
(ก) TSP, และ (ข) VRP

ปัญหาการจัดเส้นทางเดินรถ (Vehicle Routing Problem; VRP) เป็นปัญหาที่ซับซ้อนขึ้นกว่าปัญหา TSP โดยมุ่งเน้นไปที่การจัดเส้นทางให้ยานพาหนะขนส่งมากกว่าหนึ่งเส้นทาง โดยรถต้องกลับมายังจุดเริ่มต้นในขณะที่ใช้ระยะทางในการเดินทาง หรือมีค่าใช้จ่ายจากการเดินทางน้อยที่สุด ดัง

แสดงในรูปที่ 2-1(ข) เมื่อจุดสีดำ และจุดสีขาว คือ ตำแหน่งของท่ารถ (Depot) และตำแหน่งของลูกค้า ตามลำดับ ทั้งนี้ VRP อาจมีเงื่อนไขเพิ่มเติมที่เกี่ยวข้องกับข้อจำกัดด้านยานพาหนะ, ด้านเวลา, และด้านกฎหมาย เช่น การมีจำนวนยานพาหนะที่จำกัด, เวลาในการให้บริการ, และข้อจำกัดของระยะเวลาการทำงานของพนักงาน เป็นต้น (Afshar-Nadjafi and Afshar-Nadjafi, 2017, Hornstra et al., 2020, Song et al., 2020)

เนื่องจากสมมติฐานของปัญหา SDPD ในงานวิจัยนี้ กำหนดให้มีรถใช้งานเพียงคันเดียว ดังนั้น ปัญหา SDPD จึงประกอบด้วยปัญหา TSP หลายปัญหา ที่มีการปล่อยรถต่อเนื่องกันที่เวลาใดเวลาหนึ่ง โดยในแต่ละรอบของเที่ยวรถประกอบด้วย ตำแหน่งรับ และตำแหน่งส่งพัสดุ ซึ่งกระจายตัวออกไปบนแผนที่ที่ทำการศึกษา

พิจารณาตัวอย่างคำตอบของปัญหา SDPD ที่มี 5 รอบรถต่อระยะเวลาทำการ 10 ชั่วโมง ดังแสดงในรูปที่ 2-2 เมื่อจุดสีขาว, จุดสีดำ, และจุดสีเทา คือ ตำแหน่งท่ารถ, ตำแหน่งรับพัสดุ, และตำแหน่งส่งพัสดุ ตามลำดับ สังเกตเห็นว่าคำตอบนี้ประกอบไปด้วยคำตอบของปัญหา TSP จำนวน 2 คำตอบ ซึ่งเส้นทางสีน้ำเงิน และเส้นทางสีแดง คือ คำตอบของปัญหา TSP ลำดับที่หนึ่ง และลำดับที่สอง ตามลำดับ

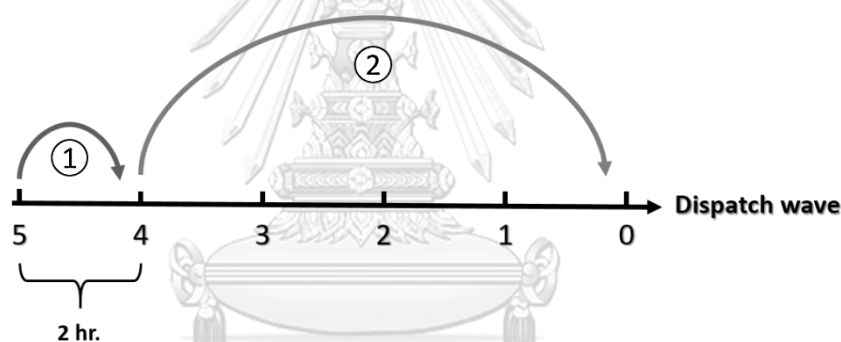


รูปที่ 2-2 ตัวอย่างคำตอบของปัญหา SDPD

หากพิจารณาการปล่อยรถ (Dispatch wave), ระยะเวลาเดินทางของรอบรถ (Dispatch duration), และรอบการกลับมาของรถ (Return wave) พบว่าทั้ง 2 เส้นทางเดินรถมีการปล่อยรถต่อเนื่องกันที่รอบรถ 4 ดังแสดงในตารางที่ 2-1 โดยสามารถจำลองแผนการเดินทางได้ด้วยรูปที่ 2-3 เมื่อลูกศรที่มีหมายเลขหนึ่งกำกับ และลูกศรที่มีหมายเลขสองกำกับ คือ เส้นทางเดินรถลำดับที่หนึ่งและลำดับที่สอง ตามลำดับ

ตารางที่ 2-1 รายละเอียดของเส้นทางเดินรถในตัวอย่างคำตอบสำหรับปัญหา SDPD

Tour	Dispatch wave	Duration		Return wave
		hr.	wave	
1	5	1.786	0.893	4
2	4	7.772	3.886	0



รูปที่ 2-3 แผนการเดินทางของตัวอย่างคำตอบสำหรับปัญหา SDPD

นอกจากนี้ ปัญหา VRP เป็นปัญหาที่ต้องใช้ระยะเวลาในการหาคำตอบด้วยวิธีแม่นยำตรง (Optimization) ที่ยาวนาน เนื่องจากปัญหา VRP มีความซับซ้อนแบบ NP-hard (Pureza et al., 2012) การใช้วิธีทางฮิวริสติกส์ (Heuristics) เพื่อหาคำตอบที่มีคุณภาพภายในระยะเวลาอันสั้น หรือภายใต้ระยะเวลาจำกัด (Good-enough and fast-enough solution) จึงเป็นวิธีที่ได้รับความนิยมมากกว่าการหาคำตอบที่ดีที่สุด (Gu et al., 2019, Martins et al., 2020, Song et al., 2020) โดยวิธีการหาคำตอบโดยทั่วไปของฮิวริสติกสามารถแบ่งออกได้เป็น 2 ขั้นตอน คือ การสร้างคำตอบเบื้องต้น และการปรับปรุงคำตอบ

2.1.1 การสร้างคำตอบเบื้องต้น (Constructive heuristics)

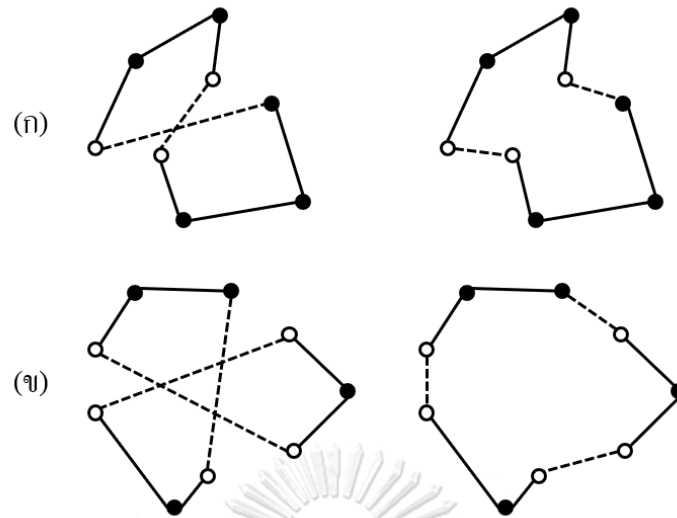
การสร้างคำตอบเบื้องต้นเป็นการสร้างคำตอบใด ๆ ที่เป็นไปได้ภายใต้ข้อจำกัด และเงื่อนไขของปัญหา ด้วยนโยบายบางประการ ยกตัวอย่างเช่น Nearest Neighbor (Sheridan et al., 2013), Clarke-Wright Savings Heuristic (Clarke and Wright, 2015), และ Tour-Partitioning Heuristics (Lewis and Sexton, 2007) เป็นต้น

สำหรับปัญหา SDPD เราอาจสร้างคำตอบเบื้องต้น ด้วยการนำคำสั่งของลูกค้าที่มีค่าตอบแทนสูงสุด มาจัดเส้นทางก่อนคำสั่งของลูกค้าที่มีค่าตอบแทนต่ำกว่า ในตำแหน่งที่เป็นไปตามข้อกำหนดของปัญหา จนไม่สามารถเพิ่มคำสั่งใด ๆ ได้อีก

2.1.2 การปรับปรุงคำตอบ (Improvement heuristics)

การปรับปรุงคำตอบเป็นวิธีการปรับปรุงคำตอบเบื้องต้น เพื่อให้เกิดคำตอบที่มีค่า Objective ที่ดีขึ้น ด้วยนโยบายบางประการ ซึ่งเราสามารถแบ่งวิธีการปรับปรุงคำตอบออกได้เป็น 2 แนวทางหลัก คือ การปรับปรุงคำตอบภายในเส้นทางเดียวกัน และการปรับปรุงคำตอบจากคนละเส้นทาง

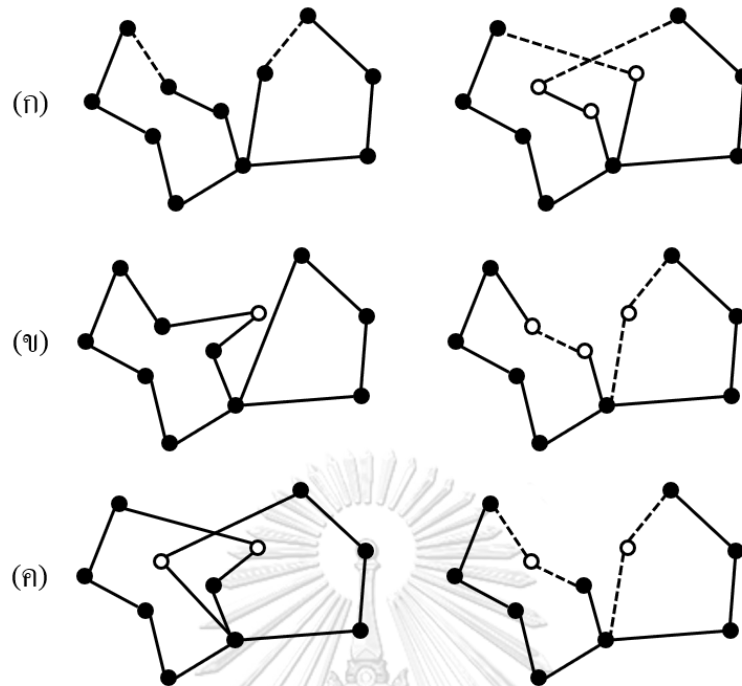
การปรับปรุงคำตอบภายในเส้นทางเดียวกัน (Intra-route improvement) คือ การสลับลำดับงานภายในเส้นทางเดินรถเส้นใดเส้นหนึ่ง เพื่อให้เกิดคำตอบที่มีค่า Objective ที่ดีขึ้น ยกตัวอย่างเช่น การสลับลำดับของลูกค้าภายในเส้นทางเดินรถเดียวกัน จำนวน 1 คู่ หรือ 2-OPT (Hougardy et al., 2020) ดังแสดงตัวอย่างในรูปที่ 2-4(ก) และการสลับลำดับของลูกค้าภายในเส้นทางเดินรถเดียวกัน จำนวน 2 คู่ หรือ 3-OPT (Mahi et al., 2015) ดังแสดงตัวอย่างในรูปที่ 2-4(ข) เมื่อจุดสีขาว, จุดสีดำ, และเส้นประ คือ ตำแหน่งลูกค้าที่ถูกสลับลำดับ, ตำแหน่งลูกค้ารายอื่น, และเส้นทางที่เกิดการเปลี่ยนแปลง ตามลำดับ



รูปที่ 2-4 การปรับปรุงคำตอบภายในเส้นทางเดียวกัน (Intra-route Improvements)

(ก) 2-OPT และ (ข) 3-OPT

การปรับปรุงคำตอบจากคนละเส้นทาง (Inter-route improvement) เป็นการสลับลำดับงาน หรือการย้ายงานระหว่างเส้นทางเดินรถคู่ใดคู่หนึ่ง เพื่อให้เกิดคำตอบที่มีค่า Objective ที่พัฒนาขึ้น ยกตัวอย่างเช่น การสลับเส้นทางจำนวน 2 เส้นทาง ระหว่างเส้นทางเดินรถ (String Cross) ดังแสดงตัวอย่างในรูปที่ 2-5(ก), การย้ายงานจำนวน 1 งานระหว่างเส้นทางเดินรถคู่ใดคู่หนึ่ง (String Relocation) ดังแสดงตัวอย่างในรูปที่ 2-5(ข) และการย้ายงานจำนวน 2 งานระหว่างเส้นทางเดินรถคู่ใดคู่หนึ่ง (SWAP) ดังแสดงตัวอย่างในรูปที่ 2-5(ค) เมื่อจุดสีขาว, จุดสีดำ, และเส้นประ คือ ตำแหน่งลูกค้าที่ถูกสลับลำดับ, ตำแหน่งลูกค้ารายอื่น, และเส้นทางที่เกิดการเปลี่ยนแปลง ตามลำดับ



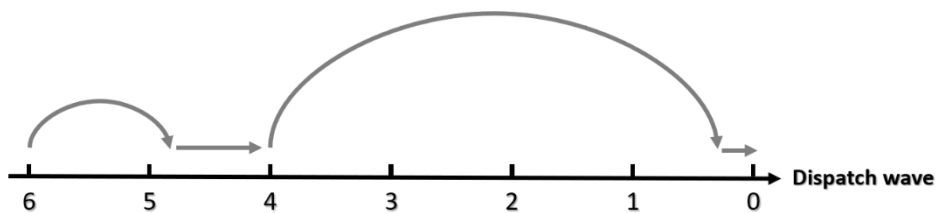
รูปที่ 2-5 การปรับปรุงคำตอบจากคนละเส้นทาง (Inter-route Improvements)

(ก) String Cross, (ข) String Relocation และ (ค) SWAP

2.2 Dynamic Dispatch Waves Problem (DDWP)

ปัญหา DDWP เป็นปัญหา VRP ที่คำสั่งของลูกค้าปรากฏขึ้นอย่างพลวัตตลอดวันทำการ และระยะเวลาทำการถูกแบ่งออกเป็นรอบรถ (Dispatch waves) จำนวนหนึ่ง โดยในแต่ละรอบรถผู้วางแผนจะทราบข้อมูลคำสั่งของลูกค้าที่ปรากฏขึ้นแล้ว (Open orders) และความพร้อมใช้งานของรถ (Vehicle availability) ด้วยเหตุนี้ รอบรถจึงเปรียบเสมือนจุดตัดสินใจสำหรับการเลือกคำสั่งซื้อบางส่วน (Subset of open orders) เพื่อนำมาจัดตารางการปล่อยรถในเวลาทำการที่หลงเหลืออยู่ โดยมีวัตถุประสงค์เพื่อลดค่าคาดหวังของค่าใช้จ่ายจากการใช้รถ และค่าคาดหวังของค่าเสียโอกาสจากการสูญงาน (Klapp et al., 2018)

ตัวอย่างการปล่อยรถของปัญหา DDWP สามารถแสดงได้ด้วยรูปที่ 2-6 เมื่อการปล่อยครั้งแรกเกิดขึ้นที่รอบรถที่ 6 และรถพร้อมกลับมาใช้งานอีกครั้งในรอบรถที่ 4 ส่วนการปล่อยครั้งที่สองเกิดขึ้นที่รอบรถที่ 4 และรถพร้อมกลับมาใช้งานอีกครั้งในรอบรถที่ 0 หรือเมื่อสิ้นสุดเวลาทำการ



รูปที่ 2-6 ตัวอย่างการปล่อยรถในปัญหา DDWP

ปัญหา DDWP มีการถ่วงดุล (Trade-offs) 2 ประการ ได้แก่ การถ่วงดุลระหว่างการชะลอปล่อยรถกับการปล่อยรถ และการถ่วงดุลระหว่างการให้บริการลูกค้าจำนวนมากต่อหนึ่งเที่ยวรถกับการให้บริการลูกค้าน้อยต่อหนึ่งเที่ยวรถ (Klapp et al., 2018)

2.2.1 การถ่วงดุลของการปล่อยรถของปัญหา DDWP

การชะลอเพื่อปล่อยรถในรอบถัดไปเป็นการเพิ่มโอกาสในการรับคำสั่งใหม่ที่อาจเกิดขึ้นระหว่างการชะลอ ซึ่งคำสั่งใหม่เหล่านี้ อาจทำให้ผลการดำเนินงานมีประสิทธิภาพมากกว่าการปล่อยรถในรอบปัจจุบันทันที แต่ทว่าการรออาจส่งผลให้ปริมาณงานค้างในระบบเพิ่มขึ้นจนไม่สามารถให้บริการบางคำสั่งได้ทันในวันทำการ ในทางกลับกันแม้ว่าการปล่อยรถในรอบปัจจุบันทันที อาจทำให้พลาดโอกาสในการรับคำสั่งใหม่ที่อาจเกิดขึ้นระหว่างรอบรถ แต่ทว่าการปล่อยรถในรอบปัจจุบันทันทีจะช่วยลดปริมาณงานค้างในระบบลง และสามารถให้บริการคำสั่งของลูกค้าได้มากขึ้นในวันทำการ

2.2.2 การถ่วงดุลของจำนวนลูกค้าที่ให้บริการลูกค้าต่อเที่ยวรถของปัญหา DDWP

การปล่อยรถเพื่อให้บริการลูกค้าจำนวนน้อยต่อหนึ่งเที่ยวรถทำให้สามารถนำรถกลับมาใช้ใหม่ได้เร็วกว่าการให้บริการลูกค้าจำนวนมากต่อหนึ่งเที่ยวรถ แต่ทว่าอาจมีค่าใช้จ่ายจากการใช้รถมาก (ไม่คุ้มเที่ยวรถ) ในทางกลับกันแม้ว่าการปล่อยรถเพื่อให้บริการลูกค้าจำนวนมากต่อหนึ่งเที่ยวรถจะทำให้เกิดค่าใช้จ่ายจากการใช้รถน้อยกว่า แต่อาจทำให้รถเสียเวลาในการเดินทางจนเกิดการขาดรถเป็นเวลานาน

ด้วยเหตุนี้ การหาคำตอบของปัญหา SDPD โดยไม่คำนึงถึงการถ่วงดุลทั้ง 2 ประการ อาจทำให้การดำเนินงานไม่มีประสิทธิภาพ หรืออาจส่งผลทำให้มีค่าใช้จ่ายจากการดำเนินงานสูงเกินความจำเป็น ดังนั้นเราจึงได้ทำการสร้างแบบจำลองทางคณิตศาสตร์ โดยกำหนดสมมติฐานให้ทราบเวลาการปรากฏของคำสั่งของลูกค้าทั้งหมดภายในวันทำการ (Deterministic problem) ก่อน ซึ่งจัดเป็น

ปัญหาดังกล่าวรูปแบบอย่างง่าย ที่สามารถนำมาศึกษา และพัฒนาวิธีการหาคำตอบของปัญหา DDWP รูปแบบ Stochastic ที่มีความไม่แน่นอนของคำสั่งของลูกค้าได้ในอนาคต

เนื่องจากปัญหา DDWP มีลักษณะเดียวกับปัญหา SDPD ในมุมมองของการบริหารจัดการทรัพยากรเวลา และรถที่มีอยู่จำกัด เพื่อตอบสนองต่อคำสั่งของลูกค้าที่มีความไม่แน่นอน ดังนั้นผู้วิจัยจึงได้ทำการสร้างแบบจำลองทางคณิตศาสตร์สำหรับปัญหา SDPD รูปแบบ Deterministic โดยประยุกต์จากปัญหา DDWP รูปแบบ Deterministic ได้ โดยมีรายละเอียดอยู่ในบทที่ 3

2.3 Pickup and Delivery Problem with Cross-Docking (PDPCD)

ปัญหา PDPCD เป็นปัญหา VRP รูปแบบหนึ่ง ที่รถต้องเดินทางไปรับพัสดุจากตำแหน่งหนึ่ง เพื่อส่งไปยังจุดหมายปลายทาง โดยที่รถสามารถทำหน้าที่เป็น Cross-dock ในการถ่าย, รวบรวม, และจัดส่งพัสดุในเที่ยวรถอื่น การแก้ปัญห PDPCD มีวัตถุประสงค์เพื่อค้นหาเส้นทางเดินรถที่มีค่าใช้จ่ายในการเดินทางน้อยที่สุด โดยพัสดุทั้งหมดต้องถูกรับ และส่งไปยังจุดหมายปลายทางโดยสมบูรณ์ (Santos et al., 2013)

ปัญหา SDPD ที่นำมาศึกษาในงานวิจัยนี้ ไม่มีข้อกำหนดให้ทุกเที่ยวรถต้องถ่ายพัสดุที่ท่ารถ เช่นเดียวกับปัญหา PDPCD ดังนั้นผู้วิจัยจึงสามารถสร้างแบบจำลองทางคณิตศาสตร์สำหรับปัญหา SDPD โดยประยุกต์จากปัญหา PDPCD ได้เช่นเดียวกัน

2.4 Adaptive Large Neighborhood Search (ALNS)

ALNS เป็นส่วนขยายของ LNS ที่เป็นอัลกอริทึมสำหรับหาคำตอบ โดย ALNS ประกอบด้วย Operator หลายตัว ที่มีหน้าที่ทำลาย และซ่อมแซมคำตอบ ซึ่ง Operator เหล่านี้เป็นฮิวริสติกสำหรับการสร้างคำตอบในเวลาอันสั้น

หลักการในการหาคำตอบของ ALNS คือ การสร้างคำตอบเริ่มต้น แล้วนำมาทำลาย และซ่อมแซม อย่างซ้ำ ๆ ด้วย Operator ที่กำหนดเพื่อค้นหาคำตอบที่มีค่า Objective ที่ดีขึ้น โดยความน่าจะเป็นในการเลือกใช้ Operator ต่าง ๆ จะถูกคำนวณจากผลลัพธ์ที่เกิดขึ้นระหว่างการหาคำตอบ เพื่อให้ ALNS เลือกใช้ Operator ได้อย่างมีประสิทธิภาพ และสอดคล้องกับการปรับปรุงค่า Objective ของโจทย์

นอกจากนี้ เรายังสามารถนำการเสาะหาคำตอบ (Local search) และการยอมรับคำตอบด้วย Simulated Annealing (SA) มาประกอบใน ALNS เพื่อเพิ่มประสิทธิภาพในการหาคำตอบได้เช่นกัน (Sacramento et al., 2019)

เนื่องจากมีงานวิจัยหลายงานที่พัฒนา ALNS สำหรับปัญหา PDP รูปแบบต่าง ๆ เช่นใน ปัญหา Pickup and Delivery Problem with Time Windows, Profits, and Reserved Requests (Li et al., 2016), ปัญหา Pickup and Delivery Problem with Time Windows and Scheduled Lines (Ghilas et al., 2016), และปัญหา Time-dependent Profitable Pickup and Delivery Problem with Time Windows (Sun et al., 2020) ผู้วิจัยจึงคาดว่า เราสามารถนำ ALNS มาประยุกต์ในปัญหา SDPD ได้ โดยรายละเอียดของ ALNS ที่พัฒนาขึ้นสำหรับงานวิจัยนี้ ถูกแสดงไว้ใน บทที่ 3

2.5 งานวิจัยที่เกี่ยวข้อง

2.5.1 The Dynamic Dispatch Waves Problem for same-day delivery

Klapp et al., (2018) ได้ทำการศึกษาาระบบการส่งพัสดุภายในวันทำการเดียวกัน (Same-day delivery: SDD) และสร้างแบบจำลอง Arc-based Integer Programming ของ ปัญหา DDWP เพื่อใช้ในการแก้ปัญหา SDD โดยมีวัตถุประสงค์หลักเพื่อลดค่าใช้จ่ายจากการใช้รถ และค่าเสียโอกาสจากการสูญเสียงานเมื่อหมดวันทำการ

ขั้นตอนการวิจัยเริ่มจากการใช้ Local search heuristics ในการแก้ปัญหา Deterministic DDWP ซึ่งเป็นรูปแบบอย่างง่ายของปัญหา DDWP ที่ทราบข้อมูลเวลาการเข้ามาของคำสั่งจัดส่งทั้งหมดล่วงหน้าตลอดทั้งวันทำการ จากนั้นจึงนำคำตอบของการแก้ปัญหา Deterministic DDWP มาออกแบบวิธีการหาคำตอบแบบ *a priori* ที่เป็นตารางระบุรอบรถ และเส้นทางที่ใช้เดินรถสำหรับคำสั่งของลูกค้าบางส่วนที่ทราบแล้ว แล้วจึงดำเนินการออกแบบนโยบายพลวัต (Dynamic policy) 2 รูปแบบ ได้แก่

i. *A-priori based rollout policy*

นโยบายนี้จะทำการสร้างคำตอบ *a priori* ที่ดีที่สุดขึ้นมาใหม่ในทุกการเริ่มต้นของรอบรถ โดยใช้สถานะปัจจุบันของระบบที่ประกอบด้วย 1.จำนวนรอบรถที่เหลือ (w) 2.เซตคำสั่งซื้อที่ปรากฏขึ้นแล้ว (R) และ 3.เซตคำสั่งซื้อที่ยังไม่ปรากฏ (P) ในการคำนวณเพื่อให้ได้คำตอบที่ดียิ่งขึ้นสำหรับรอบรถที่เหลืออยู่ ตัวอย่างเช่น ถ้านโยบาย *a priori* เดิมเสนอว่าในรอบปัจจุบันไม่ต้องปล่อยรถ แต่นโยบาย *a priori* ใหม่เสนอว่าในรอบปัจจุบันต้องปล่อยรถ ให้ปฏิบัติตามนโยบาย *a priori* ใหม่เป็นหลัก กล่าวคือปล่อยรถไปยังคำสั่งต่าง ๆ ที่นโยบาย *a priori* ใหม่กำหนด

ii. *Greedy a-priori based policy*

นโยบายนี้อาศัยข้อได้เปรียบของความสัมพันธ์ระหว่างปัญหา DDWP และ Prize-collecting TSP (PCTSP) โดยจะเริ่มต้นจากการสร้างคำตอบ *a-priori* ที่รอบรถแรก จากนั้นใช้คำตอบนี้ในการแก้ปัญหา PCTSP ทุกการเริ่มต้นของรอบรถ กล่าวคือ ในทุกรอบรถในอนาคที่คำตอบ *a priori* เสนอการปล่อยรถ ให้คำนวณระยะเวลาสูงสุดของเที่ยวรถ โดยมีเงื่อนไขว่ารถต้องกลับมายังท่ารถก่อนหน้าเที่ยวรถถัดไปที่ถูกกำหนดด้วยคำตอบ *a priori* หรือจุดสิ้นสุดเวลาทำการ จากนั้นจึงใช้เซตคำสั่งซื้อที่ทราบแล้ว (ไม่รวมคำสั่งซื้อที่ถูกกำหนดให้ขนส่งในอนาคตด้วยคำตอบ *a priori*) เป็นข้อมูลนำเข้าในการแก้ปัญหา PCTSP เพื่อกำหนดคำสั่งของลูกค้าบางส่วนที่จะให้บริการ รวมทั้งลำดับการเดินทาง

ผลการทดลองทางคอมพิวเตอร์แสดงให้เห็นว่า นโยบายที่ดีที่สุดสามารถลดค่าใช้จ่ายรวมของคำตอบ *a priori* ลงได้ 9.1 เปอร์เซ็นต์ และเพิ่มจำนวนงานที่สามารถรับได้ในวันทำการเดียวกันอย่างมาก ซึ่งเป็นการแสดงให้เห็นถึงความสำคัญของการเพิ่มประสิทธิภาพของระบบการส่งพัสดุภายในวันทำการเดียวกันด้วยนโยบายแบบพลวัต

การเพิ่มประสิทธิภาพของระบบการส่งพัสดุภายในวันทำการเดียวกัน (Same-day Delivery) ดังกล่าว ทำให้เกิดแรงบรรดาลใจในการศึกษาการรับและส่งพัสดุภายในวันทำการเดียวกัน (Same-day Pickup and Delivery) ในงานวิจัยนี้ โดยผู้วิจัยได้ทำการประยุกต์ปัญหาการจัดการจราจรแบบพลวัต (DDWP) ในการสร้างแบบจำลองทางคณิตศาสตร์ และพัฒนาวิธีการฮิวริสติกเพื่อเพิ่มกำไรสุทธิจากการดำเนินงาน ซึ่งเป็น Duality ของแบบจำลองในงานวิจัยข้างต้น

2.5.2 The Pickup and Delivery Problem with Cross-Docking

Santos et al., (2013) ได้ทำการศึกษาปัญหา Pickup and Delivery Problem with Cross-Docking (PDPCD) ซึ่งเป็นปัญหาการรับและส่งพัสดุแบบมี Cross-Dock ที่สามารถเลี่ยงการถ่ายพัสดุที่ Cross-Dock เพื่อลดค่าเดินทาง และเพิ่มพื้นที่ในการใช้ประโยชน์ของ Cross-Dock การสร้าง Integer Programming และ Branch-and-price algorithm สำหรับปัญหา PDPCD ทำให้ผลลัพธ์ค่าใช้จ่ายจากการเดินทางลดลงอย่างน้อย 3.3 เปอร์เซ็นต์ของผลลัพธ์ที่ได้จากปัญหา VRPCD ซึ่งในบางครั้งสามารถลดได้ถึง 7.1 เปอร์เซ็นต์

ผลการวิจัยของ Santos et al., (2013) ชำรงต้นทำให้เห็นแนวโน้มในการลดค่าเดินรถ หากการรับและส่งพัสดุภายในวันทำการเดียวกันที่นำมาศึกษาในงานวิจัยนี้ หากไม่มีข้อกำหนดให้ทุกเที่ยวรถต้องถ่ายพัสดุที่ Cross-Dock

2.5.3 Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests

Li et al., (2016) ได้ศึกษาปัญหาการรับและส่งพัสดุที่มีข้อจำกัดเรื่องกรอบเวลา, ผลกำไร, และคำสั่งที่ถูกจองล่วงหน้า (Pickup and Delivery Problem with Time Windows, Profits, and Reserved Requests: PDPTWPR) ซึ่งเป็นปัญหาการจัดเส้นทางเดินรถที่ต้องรับและส่งพัสดุตามคำสั่งของลูกค้าในขณะที่มีจำนวนรถจำกัด โดยวัตถุประสงค์ของงานวิจัยมุ่งเน้นไปที่การพัฒนา Mixed-Integer Linear Programming (MILP) และ Adaptive Large Neighborhood Search (ALNS) เพื่อใช้เป็นวิธีฮิวริสติกสำหรับหาคำตอบของปัญหา PDPTWPR

MILP และ ALNS ได้ถูกประเมินด้วยตัวอย่างเซตคำสั่งของลูกค้าที่ประกอบด้วยคำสั่งของลูกค้าจำนวน 10 ถึง 100 คำสั่ง จำนวน 54 ตัวอย่าง โดยได้ผลสรุปว่า คือ ALNS เป็นวิธีการหาคำตอบของปัญหา PDPTWPR ที่ดีกว่าการหาคำตอบที่ดีที่สุดด้วย MILP ทั้งในด้านคุณภาพของคำตอบ และเวลาที่ใช้ในการหาคำตอบ

ทั้งนี้จะเห็นว่าปัญหา PDPTWPR มีความคล้ายคลึงกับปัญหา SDPD ที่สนใจศึกษาในงานวิจัยนี้ เนื่องจากปัญหา SDPD มีการรับ และส่งพัสดุที่มีข้อจำกัดเรื่องกรอบเวลา เช่นเดียวกับปัญหา PDPTWPR

บทที่ 3

แนวทางการแก้ปัญหา

3.1 ลักษณะของปัญหา

ปัญหา SDPD เป็นปัญหา PDP ที่มีค่าตอบแทน และความไม่แน่นอนของคำสั่งของลูกค้า กล่าวคือ คำสั่งของลูกค้าปรากฏขึ้นอย่างสม่ำเสมอตลอดเวลาทำการ โดยในแต่ละคำสั่งของลูกค้าจะประกอบด้วยตำแหน่งรับพัสดุ (Pickup node) และตำแหน่งส่งพัสดุ (Delivery node) ที่รถต้องเดินทางไปรับ และส่งพัสดุให้ทันภายในวันทำการจึงจะได้รับค่าตอบแทนของคำสั่งนั้น ทั้งนี้ผลกำไรที่เกิดขึ้นทั้งหมด คือ ค่าตอบแทนที่เกิดขึ้นจากการปฏิบัติตามคำสั่งของลูกค้าให้สำเร็จภายในเวลาทำการ หักออกด้วยค่าใช้จ่ายจากการเดินทางที่ขึ้นกับระยะเวลาเดินทาง

ในงานวิจัยฉบับนี้ ผู้วิจัยยังได้กำหนดสมมติฐานให้ใช้นโยบายการปล่อยรถแบบรอบ (Wave-based dispatching) กล่าวคือ วันทำการจะถูกแบ่งออกเป็นรอบรถจำนวนหนึ่ง ซึ่งในแต่ละรอบรถผู้ดำเนินงานจะต้องเลือกให้บริการคำสั่งของลูกค้าที่ปรากฏขึ้นแล้ว (Open orders) แล้วจึงสร้างเส้นทางการเดินทางสำหรับรอบรถนั้น ๆ โดยมุ่งหวังให้มีกำไรสุทธิจากแผนการเดินทางสูงที่สุด ทั้งนี้ผู้วิจัยได้ทำการสร้างแบบจำลองทางคณิตศาสตร์สำหรับปัญหา SDPD โดยอ้างอิงจากปัญหา DDWP และปัญหา PDP

ปัญหา SDPD ที่ใช้นโยบายการปล่อยรถแบบเป็นรอบมีการถ่วงดุล (Trade-offs) เช่นเดียวกับกับปัญหา DDWP ดังที่ Klapp et al., (2018) ได้กล่าวไว้ในงานวิจัย ซึ่งประกอบไปด้วยการถ่วงดุลระหว่างการชะลอปล่อยรถกับการปล่อยรถ และการถ่วงดุลระหว่างการให้บริการลูกค้าจำนวนมากต่อหนึ่งเที่ยวรถกับการให้บริการลูกค้าน้อยต่อหนึ่งเที่ยวรถ

การถ่วงดุลของการปล่อยรถ

การชะลอเพื่อปล่อยรถในรอบรถถัดไปเป็นการเพิ่มโอกาสรับคำสั่งใหม่ที่เกิดขึ้นระหว่างการชะลอ ซึ่งคำสั่งใหม่เหล่านี้ อาจทำให้ผลการดำเนินงานมีประสิทธิภาพมากกว่าการปล่อยรถในรอบปัจจุบันทันที แต่ทว่าการรออาจส่งผลให้ปริมาณงานค้างในระบบเพิ่มขึ้นจนไม่สามารถให้บริการบางคำสั่งได้ทันในวันทำการ ในทางกลับกัน แม้ว่าการปล่อยรถในรอบปัจจุบันทันที อาจทำให้พลาดโอกาสในการรับคำสั่งใหม่ที่เกิดขึ้นระหว่างรอบรถ แต่ทว่าการปล่อยรถในรอบปัจจุบันทันทีจะช่วยลดปริมาณงานค้างในระบบลง และสามารถให้บริการคำสั่งของลูกค้าได้มากขึ้นในวันทำการ

การถ่วงดุลของจำนวนลูกค้าที่ให้บริการลูกค้าต่อเที่ยวรถ

การปล่อยรถเพื่อให้บริการลูกค้าจำนวนน้อยต่อหนึ่งเที่ยวรถทำให้สามารถนำรถกลับมาใช้ใหม่ได้เร็วกว่าการให้บริการลูกค้าจำนวนมากต่อหนึ่งเที่ยวรถ แต่ทว่าอาจมีค่าใช้จ่ายจากการใช้รถมาก (ไม่คุ้มเที่ยวรถ) ในทางกลับกันแม้ว่าการปล่อยรถเพื่อให้บริการลูกค้าจำนวนมากต่อหนึ่งเที่ยวรถจะทำให้เกิดค่าใช้จ่ายจากการใช้รถน้อยกว่า แต่อาจทำให้รถเสียเวลาในการเดินทางจนเกิดการขาดรถเป็นเวลานาน

ด้วยเหตุข้างต้นการหาคำตอบของปัญหา SDPD โดยไม่คำนึงถึงการถ่วงดุลดังกล่าว อาจทำให้การดำเนินงานไม่มีประสิทธิภาพ หรืออาจส่งผลทำให้มีค่าใช้จ่ายจากการดำเนินงานสูงเกินความจำเป็น ผู้วิจัยได้ทำการสร้างแบบจำลองทางคณิตศาสตร์สำหรับปัญหาดังกล่าว โดยกำหนดสมมติฐานให้ทราบเวลาการปรากฏของคำสั่งของลูกค้าทั้งหมดภายในวันทำการ (Deterministic problem) เพื่อนำมาทดลองหาคำตอบที่ดีที่สุด (Optimization) กับปัญหาขนาดต่าง ๆ และใช้เป็นเกณฑ์ในการพัฒนาฮิวริสติก (Heuristic) ในลำดับถัดไป

3.2 สมมติฐานของปัญหา

แบบจำลองทางคณิตศาสตร์ และฮิวริสติกสำหรับปัญหา SDPD แบบ Deterministic ถูกพัฒนาขึ้นโดยตั้งอยู่บนสมมติฐานที่มีรายละเอียดดังต่อไปนี้

- เวลาการปรากฏของคำสั่งของลูกค้ามีการกระจายตัวของความน่าจะเป็นแบบมีรูปแบบแน่นอน (Discrete uniform distribution)
- ตำแหน่งของคำสั่งของลูกค้ามีการกระจายตัวของความน่าจะเป็นแบบมีรูปแบบแน่นอน
- ท่ารถเดียว (Single depot)
- รถคันเดียว (Single vehicle)
- รถเริ่มต้นและสิ้นสุดการเดินทางที่ท่ารถเท่านั้น
- รถสามารถออกเดินทางจากท่ารถได้ ณ รอบรถใดรอบรถหนึ่ง เพื่อให้บริการคำสั่งของลูกค้าที่ปรากฏแล้ว (Open orders) เท่านั้น
- รถต้องกลับมายังท่ารถก่อนหมดเวลาทำการเสมอ
- ไม่คำนึงถึงเวลาที่ใช้ในการขนพัสดุขึ้นรถ (Loading time) และเวลาที่ใช้ในการถ่ายพัสดุดลงจากรถ (Unloading time)
- ไม่คำนึงถึงความจุของรถ

3.3 แนวทางการดำเนินงาน

แนวทางการดำเนินงานของงานวิจัยนี้ แบ่งออกเป็น 3 ขั้นตอน ได้แก่ การตั้งค่าของปัญหา SDPD, การสร้างแบบจำลองทางคณิตศาสตร์สำหรับปัญหา SDPD, และการพัฒนาฮิวริสติกสำหรับปัญหา SDPD

3.3.1 การตั้งค่าของปัญหา SDPD

ช่วงเวลาทำการ

ช่วงเวลาทำการในแต่ละวันถูกแบ่งออกเป็นรอบรถจำนวน ω รอบที่มีระยะเวลาห่างกันเท่ากับ ρ โดยที่หมายเลขของรอบรถ คือ จำนวนรอบรถที่หลงเหลืออยู่ในวันทำการ (Remaining waves) ซึ่งการปล่อยรถในแต่ละรอบจะเกิดขึ้นได้ เมื่อมีรถอยู่ที่ท่ารถเท่านั้น ดังแสดงในรูปที่ 3-1



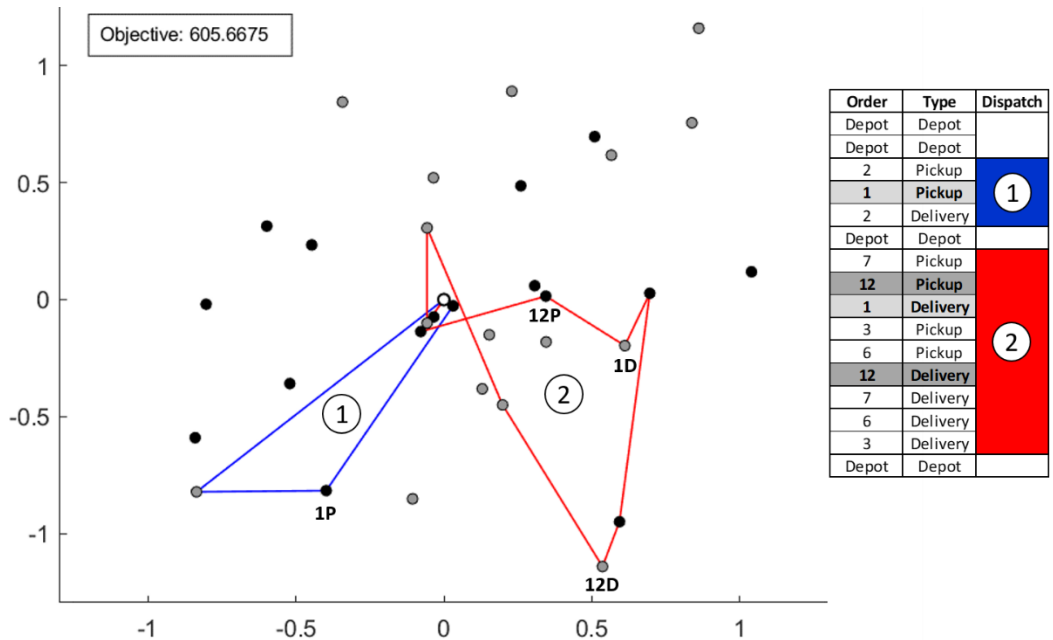
รูปที่ 3-1 ช่วงเวลาทำการของปัญหา SDPD

คำสั่งของลูกค้า และข้อกำหนดในการจัดเส้นทาง

คำสั่งของลูกค้าที่เป็นไปได้ทั้งหมดในวันทำการมีจำนวน n คำสั่ง โดยในแต่ละคำสั่งจะประกอบไปด้วยตำแหน่งรับพัสดุ (Pickup node) และตำแหน่งส่งพัสดุ (Delivery node) ที่เป็นอิสระต่อกัน

สำหรับข้อกำหนดในการจัดเส้นทางหลัก ๆ คือ ต้องจัดเส้นทางให้รถผ่านตำแหน่งรับพัสดุก่อนตำแหน่งส่งพัสดุเสมอ โดยรอบรถจะมีจุดเริ่มต้น และจุดสิ้นสุดที่ท่ารถ (Depot node : \emptyset) ตัวอย่างการจัดเส้นทางของคำสั่งลำดับที่ 1 ที่มีตำแหน่งรับพัสดุ และตำแหน่งส่งพัสดุอยู่คนละเที่ยวรถกัน และคำสั่งลำดับที่ 12 ที่มีตำแหน่งรับพัสดุ และตำแหน่งส่งพัสดุอยู่เที่ยวรถเดียวกัน สามารถแสดงได้ด้วยรูปที่ 3-2 เมื่อจุดสีขาวยังคือ ตำแหน่งท่ารถ ส่วนจุด 1P, 1D, 12P, และ 12D คือ ตำแหน่งรับพัสดุ

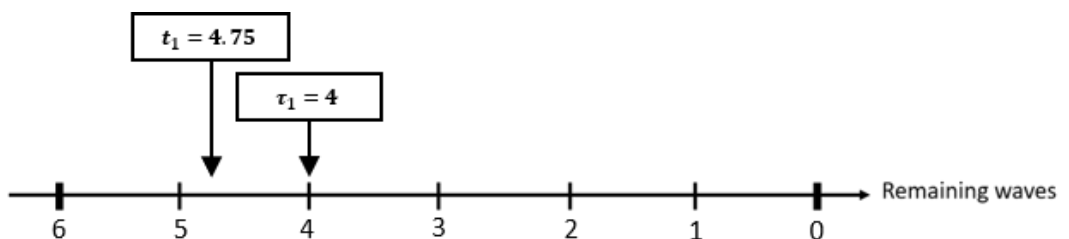
ของคำสั่งลำดับที่ 1, ตำแหน่งส่งพัสดุของคำสั่งลำดับที่ 1, ตำแหน่งรับพัสดุของคำสั่งลำดับที่ 12, และตำแหน่งส่งพัสดุของคำสั่งลำดับที่ 12 ตามลำดับ



รูปที่ 3-2 ข้อกำหนดในการจัดเส้นทางของปัญหา SDPD

เวลาการปรากฏ (Arrival times) และรอบการปรากฏ (Arrival waves)

เวลาการปรากฏของคำสั่ง i (t_i) คือ เวลาที่คำสั่ง i ปรากฏขึ้นในระบบจากการนัดหมายของลูกค้า ส่วนรอบการปรากฏของคำสั่ง i (τ_i) คือ รอบรถแรกที่สามารถนำคำสั่ง i ไปประกอบการตัดสินใจได้ กล่าวคือเป็นรอบรถถัดไปที่อยู่ใกล้ t_i มากที่สุด ทั้งนี้ตัวอย่างที่คำสั่งลำดับที่ 1 มีเวลาการปรากฏอยู่ระหว่างรอบรถที่ 4 กับรอบรถที่ 5 ($t_1 = 4.75$) และมีรอบการปรากฏ คือ รอบรถที่ 4 ($\tau_1 = 4$) สามารถแสดงได้ด้วยรูปที่ 3-3



รูปที่ 3-3 ตัวอย่างเวลาการปรากฏ และรอบการปรากฏของคำสั่ง

ค่าใช้จ่ายในการเดินทาง และค่าตอบแทน (Travel costs and Order prizes)

ค่าใช้จ่ายในการเดินทางจากตำแหน่ง i และตำแหน่ง j (c_{ij}) มีอัตราส่วนหนึ่งต่อหนึ่งกับระยะเวลาที่ใช้ในการเดินทางระหว่างสองตำแหน่ง (t_{ij}) ซึ่งระยะเวลาเดินทางก็มีอัตราส่วนหนึ่งต่อหนึ่งกับระยะห่างระหว่างสองตำแหน่งเช่นเดียวกัน ดังนั้น c_{ij} จึงสามารถคำนวณได้จากระยะห่างระหว่างตำแหน่ง i และตำแหน่ง j ตามสมการที่ (3.1) เมื่อ (x_i, y_i) คือ คู่อันดับแสดงตำแหน่งของตำแหน่ง i และ (x_j, y_j) คือ คู่อันดับแสดงตำแหน่งของตำแหน่ง j บนระบบแกนพิกัดฉาก

$$c_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3.1)$$

ค่าตอบแทน คือ รายรับสมมติของคำสั่งที่กำหนดขึ้นเพื่อให้มีความจำเป็นในการให้บริการ ซึ่งค่าตอบแทนจะได้รับก็ต่อเมื่อมีการส่งพัสดุตามคำสั่งได้ภายในเวลาทำการ ค่าตอบแทนของคำสั่ง r (P_r) ถูกกำหนดให้แปรผันตรงกับค่าใช้จ่ายในการเดินทาง จากตำแหน่งรับพัสดุไปยังตำแหน่งส่งพัสดุของคำสั่งนั้น ซึ่งสามารถคำนวณได้จากสมการ (3.2) เมื่อ $\alpha > 0$ เป็นค่าคงที่สำหรับเพิ่มมูลค่าของค่าตอบแทน และ $|a_{p_r d_r}|$ คือ ระยะห่างระหว่างตำแหน่งรับ และตำแหน่งส่งพัสดุของคำสั่ง r

$$P_r = \alpha c_{p_r d_r} \quad (3.2)$$

3.3.2 การสร้างแบบจำลองทางคณิตศาสตร์สำหรับปัญหา SDPD แบบ Deterministic

แบบจำลองทางคณิตศาสตร์สำหรับปัญหา SDPD รูปแบบ Deterministic ถูกสร้างขึ้นโดยอ้างอิงจากปัญหา DDWP รูปแบบ Deterministic และปัญหา PDP ซึ่งมีฟังก์ชันจุดประสงค์ (Objective function) เป็นไปเพื่อการทำให้กำไรสุทธิจากการดำเนินงานสูงที่สุด

เซต และพารามิเตอร์

แบบจำลองทางคณิตศาสตร์ที่สร้างขึ้นประกอบด้วยเซต และพารามิเตอร์ที่สามารถแบ่งกลุ่มได้เป็น เซต และพารามิเตอร์ทั่วไป ดังแสดงในตารางที่ 3-1 กับเซต และพารามิเตอร์ของรอบรถ ดังแสดงในตารางที่ 3-2

ตารางที่ 3-1 เซต และพารามิเตอร์ทั่วไปในแบบจำลองทางคณิตศาสตร์

เซตของรอบรถ	$W := \{ \omega, \omega - 1, \omega - 2, \dots, 2, 1, 0 \}$
เซตของคำสั่งลูกค้า	$R := \{ 1, 2, \dots, n \}$
เซตของตำแหน่งรับพัสดุ	$P := \{ p_i \mid i \in R \}$
เซตของตำแหน่งส่งพัสดุ	$D := \{ d_i \mid i \in R \}$
เซตของเส้นทางทั้งหมด	$A := \{ a_{ij} \mid \forall i, j \in P \cup D \}$
รอบรถที่ช้าที่สุดในการเดินทางไปตำแหน่ง $m \in P \cup D$	$w_m = \lceil 2t_{(0,m)} / \ell \rceil$
รอบรถที่ช้าที่สุดที่สามารถใช้เส้นทาง $a_{ij} \in A$	$w_{ij}^a = \lceil (t_{0i} + t_{ij} + t_{0j}) / \ell \rceil$
รอบรถที่เร็วที่สุดที่สามารถใช้เส้นทาง $a_{ij} \in A$	$w_{ij}^b = \min(\tau_i, \tau_j) > 0$

ตารางที่ 3-2 เซต และพารามิเตอร์ของรอบรถในแบบจำลองทางคณิตศาสตร์

เซตของตำแหน่งรับพัสดุในรอบรถ $w \in W$	$N_w^p := \{ p_i \mid i \in N : a_{p_i} \leq w \leq \tau_i \}$
เซตของตำแหน่งส่งพัสดุในรอบรถ $w \in W$	$N_w^d := \{ d_i \mid i \in N : a_{d_i} \leq w \leq \tau_i \}$
เซตของตำแหน่งทั้งหมดในรอบรถ $w \in W$	$N_w := N_w^p \cup N_w^d$
เซตของเส้นทางในรอบรถ $w \in W$	$A_w := \{ a_{ij} \in A : w_{ij}^a \leq w \leq w_{ij}^b \}$

ตัวแปรตัดสินใจ (Decision variables)

ตัวแปรตัดสินใจในแบบจำลองทางคณิตศาสตร์นี้ประกอบไปด้วย ตัวแปรตัดสินใจสำหรับมอบหมายคำสั่ง, ตัวแปรตัดสินใจสำหรับเลือกเส้นทาง, ตัวแปรตัดสินใจสำหรับควบคุมการเดินทางของรถ, ตัวแปรตัดสินใจสำหรับควบคุมการชะลอของรถ, และตัวแปรตัดสินใจสำหรับกำจัดพัสดุ

ตัวแปรตัดสินใจสำหรับมอบหมายคำสั่ง

$$y_i^w \in 0, 1 \quad \forall i \in p_r \cup d_r, \forall w \in W$$

$$y_i^w = \begin{cases} 1 & ; \text{ตำแหน่ง } i \in N \text{ ถูกมอบหมายในรอบ } w \in W \\ 0 & ; \text{กรณีอื่น} \end{cases}$$

ตัวแปรตัดสินใจสำหรับเลือกเส้นทาง

$$x_a^w \in \{0, 1\} \quad \forall a \in A_w, \forall w \in W$$

$$x_a^w = \begin{cases} 1 & ; \text{ใช้เส้นทาง } a \in A_w \text{ ในรอบ } w \in W \\ 0 & ; \text{กรณีอื่น} \end{cases}$$

ตัวแปรตัดสินใจสำหรับควบคุมการเดินทางของรถ

$$v_k^w \in \{0, 1\} \quad \forall w \in W, \forall k \in W : k < w$$

$$v_k^w = \begin{cases} 1 & ; \text{ใช้เที่ยวรถที่เดินทางจากรอบ } w \in W \text{ ถึง } k \in W \\ 0 & ; \text{กรณีอื่น} \end{cases}$$

ตัวแปรตัดสินใจสำหรับควบคุมการชะลอของรถ

$$z_k \in \{0, 1\} \quad \forall k \in W : k < \omega$$

$$z_k = \begin{cases} 1 & ; \text{ไม่มีการปล่อยรถจนกระทั่งรอบ } k \in W \\ 0 & ; \text{กรณีอื่น} \end{cases}$$

ตัวแปรตัดสินใจสำหรับกำจัดตัวร่อย

$$k_i^w \in \mathbb{Z} \quad \forall w \in W, \forall i \in N_w$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$C^* = \max_{(x,y,v,z)} \sum_{r \in R} \sum_{w \in W_{p_r}}^{\tau_r} P_r y_{p_r}^w - \gamma \sum_{w \in W} \sum_{a \in A_w} t_a x_a^w$$

subject to

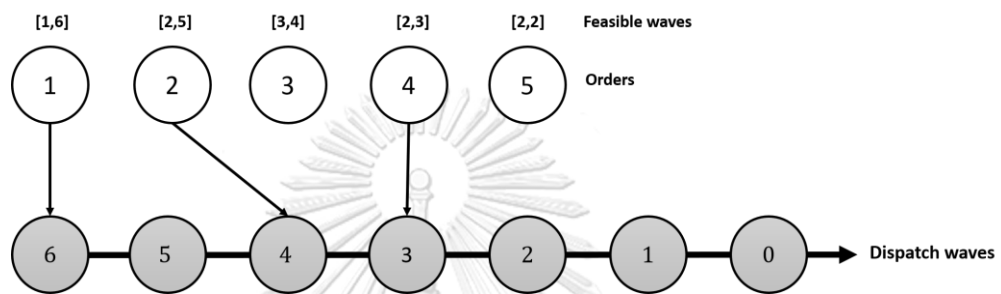
- i. $\sum_{w \in W_{p_r}}^{\tau_r} y_{p_r}^w \leq 1 \quad \forall r \in R$
- ii. $\sum_{w \in W_{p_r}}^{\tau_r} y_{p_r}^w = \sum_{w \in W_{d_r}}^{\tau_r} y_{d_r}^w \quad \forall r \in R$
- iii. $\sum_{w \in W_{d_r}: k \leq w}^{\tau_r} y_{d_r}^w \geq y_{p_r}^w \quad \forall w \in W, \forall r \in R$
- iv. $\sum_{a_{ij} \in A_w: j \in N_w} x_{ij}^w = y_i^w \quad \forall w \in W, \forall i \in N_w$
- v. $\sum_{a_{ji} \in A_w: j \in N_w} x_{ji}^w = y_i^w \quad \forall w \in W, \forall i \in N_w$
- vi. $k_j^w \geq k_i^w + x_{ij}^w - M_w(1 - x_{ij}^w) \quad \forall w \in W, \forall i, j \in N_w: i \neq j, i \neq \emptyset$
- vii. $k_j^w \geq x_{\emptyset j}^w - M_w(1 - x_{\emptyset j}^w) \quad \forall w \in W, \forall j \in N_w: j \neq \emptyset$
- viii. $k_{p_r}^w \leq k_{d_r}^w \quad \forall w \in W, \forall r \in R$
- ix. $\sum_{k < \omega} z_k + \sum_{k < \omega} v_k^\omega = 1$
- x. $\sum_{k < w} v_k^w = \sum_{k' > w} v_w^{k'} + z_w \quad \forall w \in W \setminus \{\omega\}$
- xi. $\sum_{a \in A_w} t_a x_a^w \leq \ell \sum_{k < w} (w - k) v_k^w \quad \forall w \in W$
- xii. $k_i^w \geq 0 \quad \forall w \in W, \forall i \in N_w$
- xiii. $x_a^w \in \{0, 1\} \quad \forall a \in A_w, \forall w \in W$
- xiv. $v_k^w \in \{0, 1\} \quad \forall w \in W, \forall k \in W: k < w$
- xv. $z_k \in \{0, 1\} \quad \forall k \in W: k < \omega$

คำอธิบายฟังก์ชันวัตถุประสงค์

การทำให้กำไรสุทธิจากการดำเนินงานมีค่าสูงที่สุด ซึ่งกำไรสุทธิตำหนดได้จากค่าตอบแทนทั้งหมดภายในวันทำการ ($\sum_{r \in R} \sum_{w \in W_{p_r}}^{\tau_r} P_r y_{p_r}^w$) หักออกด้วยค่าเดินทางทั้งหมดภายในวันทำการ ($\gamma \sum_{w \in W} \sum_{a \in A_w} t_a x_a^w$)

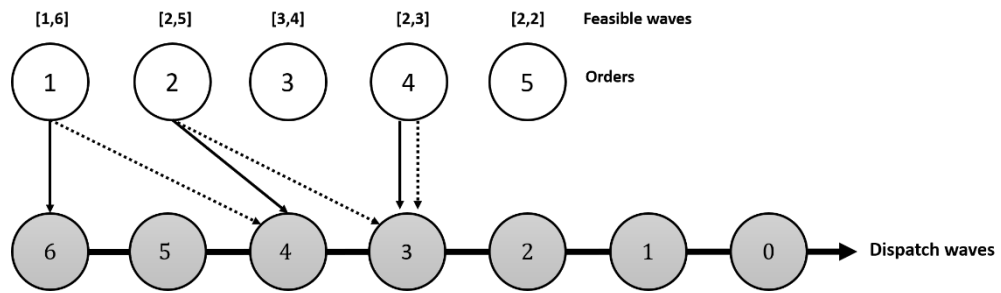
อสมการข้อจำกัดสำหรับมอบหมายงาน (อสมการที่ i. ถึง iii.)

- i. แต่ละคำสั่งของลูกค้า $r \in R$ ต้องถูกกำหนดให้อยู่ในเส้นทางเดินรถรอบใดรอบหนึ่งเพื่อรับพัสดุ ($\sum_{w \in w_{p_r}} y_{p_r}^w = 1$) หรือไม่อยู่ในเส้นทางเดินรถของรอบรถรอบใดเลย ($\sum_{w \in w_{p_r}} y_{p_r}^w = 0$) ในช่วงเวลาที่เป็นไปได้ในการเดินทางไปที่ตำแหน่งรับพัสดุ r ($[a_{p_r}, \tau_r]$) ตัวอย่างเช่น ในรูปที่ 3-4 แสดงตัวอย่างคำตอบที่รถในรอบที่ 6, 4, และ 3 มีหน้าที่รับพัสดุของคำสั่งของลูกค้าลำดับที่ 1, 2 และ 4 ตามลำดับ ($y_{p_1}^6 = y_{p_2}^4 = y_{p_4}^3 = 1$) เมื่อลูกศรเส้นทึบ คือ การมอบหมายงานรับพัสดุในรอบรถหนึ่ง ๆ



รูปที่ 3-4 ตัวอย่างการทำงานของอสมการข้อจำกัดที่ i.

- ii. แต่ละคำสั่งของลูกค้า $r \in R$ ที่ถูกกำหนดให้อยู่ในเส้นทางเดินรถของรอบรถรอบใดรอบหนึ่งเพื่อรับพัสดุ ($\sum_{w \in w_{p_r}} y_{p_r}^w = 1$) ต้องถูกกำหนดให้อยู่ในเส้นทางเดินรถของรอบรถรอบใดรอบหนึ่งเพื่อส่งพัสดุ ($\sum_{w \in w_{d_r}} y_{d_r}^w = 1$) ในช่วงเวลาที่เป็นไปได้ $[a_{d_r}, \tau_r]$ ในทางกลับกัน หากคำสั่งของลูกค้า r ไม่ถูกกำหนดให้อยู่ในเส้นทางเดินรถของรอบรถรอบใดรอบหนึ่งเพื่อรับพัสดุ ($\sum_{w \in w_{p_r}} y_{p_r}^w = 0$) คำสั่งของลูกค้า r ต้องไม่อยู่ในเส้นทางเดินรถของรอบรถรอบใดรอบหนึ่งเพื่อส่งพัสดุเช่นกัน ($\sum_{w \in w_{d_r}} y_{d_r}^w = 0$) รูปที่ 3-5 แสดงตัวอย่างคำตอบที่รถในรอบที่ 6 มีหน้าที่รับพัสดุของคำสั่งของลูกค้าลำดับที่ 1, รถในรอบที่ 4 มีหน้าที่รับพัสดุของคำสั่งของลูกค้าลำดับที่ 2 และส่งพัสดุของคำสั่งของลูกค้าลำดับที่ 1, รถในรอบที่ 3 มีหน้าที่รับพัสดุของคำสั่งของลูกค้าลำดับที่ 4 และส่งพัสดุของคำสั่งของลูกค้าลำดับที่ 2 และ 4 เมื่อลูกศรเส้นทึบ และลูกศรเส้นประ คือ การมอบหมายงานรับพัสดุในรอบรถหนึ่ง ๆ และการมอบหมายงานส่งพัสดุในรอบรถหนึ่ง ๆ ตามลำดับ



รูปที่ 3-5 ตัวอย่างการทำงานของอสมการข้อจำกัดที่ ii.

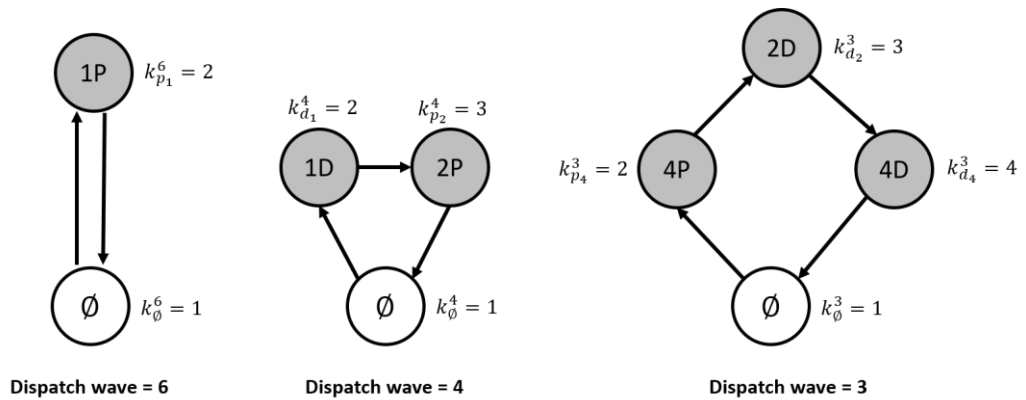
- iii. แต่ละรอบรถ $w \in W$ และแต่ละคำสั่งของลูกค้า $r \in R$ รอบรถในการส่งพัสดุของคำสั่งของลูกค้า r ต้องเกิดขึ้นภายหลัง หรือเกิดขึ้นในรอบเดียวกันกับรอบรถที่รับพัสดุของลูกค้า r เสมอ ดังแสดงตัวอย่างในรูปที่ 3-5

อสมการข้อจำกัดสำหรับกำหนดเส้นทางเดินรถ (อสมการที่ iv. ถึง v.)

- iv. แต่ละรอบรถ $w \in W$ และแต่ละตำแหน่งในเซตของตำแหน่งทั้งหมดในรอบรถ $i \in N_w$ ต้องมีการใช้เส้นทางที่เป็นไปได้ในรอบรถ w ที่มีจุดเริ่มต้นจาก i เฉพาะกรณีที่รถได้รับมอบหมายให้เดินทางไปยังตำแหน่งนั้น ($y_i^w = 1$)
- v. แต่ละรอบรถ $w \in W$ และแต่ละตำแหน่งในเซตของตำแหน่งทั้งหมดในรอบรถ $i \in N_w$ ต้องมีการใช้เส้นทางที่เป็นไปได้ในรอบรถ w ที่มีจุดสิ้นสุด คือ ตำแหน่ง i เฉพาะกรณีที่รถได้รับมอบหมายให้เดินทางไปยังตำแหน่งนั้น ($y_i^w = 1$)

อสมการข้อจำกัดสำหรับกำจัดทัวร์ย่อย (อสมการที่ vi. ถึง vii.)

- vi. อสมการนี้มีหน้าที่ควบคุมให้ตัวแปรตัดสินใจ k ของตำแหน่งที่ไม่ใช่ท่ารถ มีค่าเพิ่มขึ้นตามเส้นทางเดินรถ เพื่อป้องกันไม่ให้เกิดทัวร์ย่อยในคำตอบ ดังแสดงตัวอย่างในรูปที่ 3-6 เมื่อจุดสีขาว คือ ตำแหน่งท่ารถ ส่วนจุด 1P, 1D, 2P, 2D, 4P และ 4D คือ ตำแหน่งรับพัสดุของคำสั่งลำดับที่ 1, ตำแหน่งส่งพัสดุของคำสั่งลำดับที่ 1, ตำแหน่งรับพัสดุของคำสั่งลำดับที่ 2, ตำแหน่งส่งพัสดุของคำสั่งลำดับที่ 2, ตำแหน่งรับพัสดุของคำสั่งลำดับที่ 4, และตำแหน่งส่งพัสดุของคำสั่งลำดับที่ 4 ตามลำดับ



รูปที่ 3-6 ตัวอย่างการทำงานของอสมการข้อจำกัดที่ vi. ถึง viii.

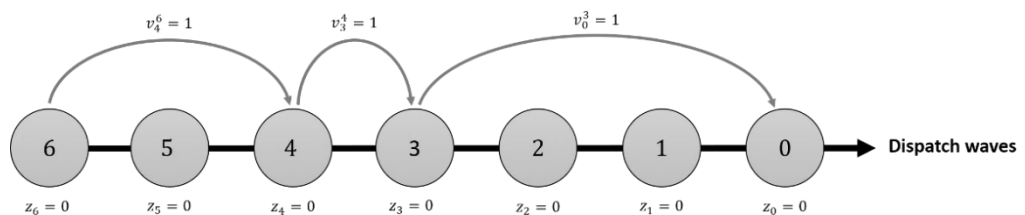
vii. อสมการนี้มีหน้าที่ควบคุมให้ตัวแปรตัดสินใจ k ของตำแหน่งท่ารถ (\emptyset) ให้มีค่าเท่ากับหนึ่ง ดังแสดงตัวอย่างในรูปที่ 3-6

อสมการข้อจำกัดสำหรับควบคุมการรับและส่งพัสดุในรอบรถเดียวกัน (อสมการที่ viii.)

viii. อสมการนี้มีหน้าที่ควบคุมให้ตัวแปรตัดสินใจ k ของตำแหน่งรับพัสดุ มีค่าน้อยกว่าค่า k ของตำแหน่งส่งพัสดุของคำสั่งลำดับเดียวกัน ดังแสดงตัวอย่างในรอบรถที่ 3 ของรูปที่ 3-6

อสมการข้อจำกัดสำหรับควบคุมการเดินทางของรถ (อสมการที่ ix. ถึง xi.)

ix. รอบรถแรกสุด (ω) มีรถหนึ่งคันที่มีสองทางเลือก ได้แก่ การปล่อยรถจากรอบรถที่ ω แล้วกลับมาพร้อมใช้งานอีกครั้งที่รอบรถที่ k ($\sum_{k < \omega} v_k^\omega = 1$) หรือ การรออยู่ที่ท่ารถตั้งแต่รอบรถที่ ω จนถึงรอบรถที่ k ($\sum_{k < \omega} z_k = 1$) สำหรับตัวอย่างที่มีการปล่อยรถจากรอบรถที่ 6 ($\omega = 6$) แล้วกลับมาที่รอบรถที่ 4 สามารถแสดงได้ด้วยรูปที่ 3-7 เมื่อรอบที่อยู่ส่วนหางของลูกศร คือ รอบที่มีการปล่อยรถ ส่วนรอบที่อยู่ส่วนหัวของลูกศร คือ รอบถัดไปที่สามารถปล่อยรถได้ (รถกลับมายังท่ารถแล้ว)



รูปที่ 3-7 ตัวอย่างการทำงานของอสมการข้อจำกัดที่ ix. ถึง xi.

- x. แต่ละรอบรถ $w \in W$ ต้องมีการไหลเข้า-ออกของรถเท่ากัน กล่าวคือ หากรถกลับมายังท่ารถที่รอบ w ให้ทำการปล่อยรถในรอบนั้นทันที ตัวอย่างที่รถกลับมายังท่ารถในรอบรถที่ 4 และรอบรถที่ 3 สามารถแสดงได้ด้วยรูปที่ 3-7 เมื่อรอบที่อยู่ส่วนทางลูกศรสีแดง คือ รอบที่มีการปล่อยรถ ส่วนรอบที่อยู่ส่วนหัวลูกศรสีแดง คือ รอบที่รถกลับมาพร้อมใช้งาน
- xi. แต่ละรอบรถ $w \in W$ ที่มีการจัดเส้นทาง ($\sum_{a \in A_w} t_a x_a^w > 0$) รถจะกลับมายังท่ารถในรอบรถถัดไป โดยใช้เวลาไม่มากกว่าสัดส่วนระหว่าง $\sum_{a \in A_w} t_a x_a^w$ และ ℓ เสมอ

อสมการข้อจำกัดสำหรับควบคุมค่าของตัวแปรตัดสินใจ (อสมการที่ xii. ถึง xv.)

- xii. อสมการนี้มีหน้าที่ควบคุมให้ตัวแปรตัดสินใจ k มีค่ามากกว่า หรือเท่ากับศูนย์เสมอ
- xiii. อสมการนี้มีหน้าที่ควบคุมให้ตัวแปรตัดสินใจ x มีค่าเท่ากับหนึ่ง หรือศูนย์เท่านั้น
- xiv. อสมการนี้มีหน้าที่ควบคุมให้ตัวแปรตัดสินใจ y มีค่าเท่ากับหนึ่ง หรือศูนย์เท่านั้น
- xv. อสมการนี้มีหน้าที่ควบคุมให้ตัวแปรตัดสินใจ z มีค่าเท่ากับหนึ่ง หรือศูนย์เท่านั้น

3.3.3 การพัฒนาฮิวริสติกสำหรับปัญหา SDPD

ปัญหา SDPD เป็นปัญหาการจัดเส้นทางเดินรถรูปแบบหนึ่ง ที่มีความซับซ้อนแบบ NP-hard (Pureza et al., 2012) ด้วยเหตุนี้การหาคำตอบที่ดีที่สุด (Optimization) จึงใช้เวลายาวนาน และไม่เหมาะสมกับปัญหาขนาดใหญ่ ผู้วิจัยจึงได้พัฒนาฮิวริสติก Adaptive Large Neighborhood Search (ALNS) สำหรับหาคำตอบของปัญหา SDPD เพื่อให้ได้คำตอบที่มีคุณภาพภายในระยะเวลาอันสั้น ทั้งนี้การพัฒนาฮิวริสติกประกอบด้วย 2 ขั้นตอน คือ การนำเข้าข้อมูลของปัญหา และการหาคำตอบของฮิวริสติก

การนำเข้าข้อมูลของปัญหา

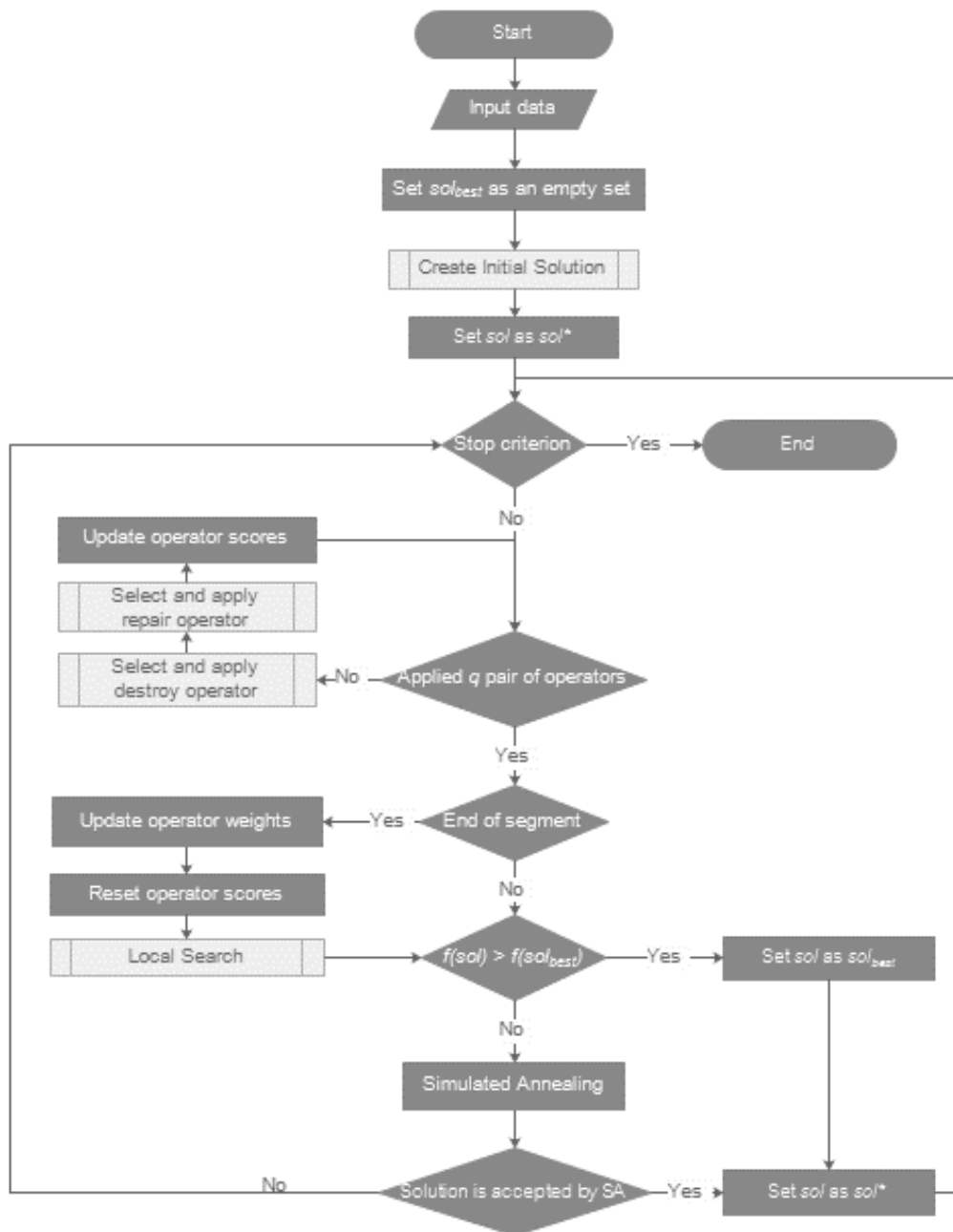
ข้อมูลนำเข้าของปัญหา SDPD ประกอบไปด้วย ลำดับคำสั่ง (Order numbers), พิกัดรับพัสดุ (Pickup coordinates), พิกัดส่งพัสดุ (Delivery coordinates), ค่าตอบแทนของคำสั่ง (Order prizes), เวลาปรากฏของคำสั่ง (Arrival time), และรอบปรากฏของคำสั่ง (Arrival wave) ตัวอย่างข้อมูลนำเข้าที่มีจำนวนคำสั่งทั้งหมดที่เป็นไปได้เท่ากับ 16 คำสั่ง และ 6 รอบรถ สามารถแสดงได้ด้วยตารางที่ 3-3

ตารางที่ 3-3 ตัวอย่างข้อมูลนำเข้าของปัญหา SDPD

Order No.	Pickup node		Delivery node		Order prize	Arrival time	Arrival wave
	X	Y	X	Y			
1	-0.3975	-0.8164	0.6115	-0.1968	118.3966	5.9546	5
2	0.0310	-0.0265	-0.8356	-0.8216	117.6104	5.8591	5
3	0.6958	0.0269	-0.0566	-0.1004	76.3094	5.7914	5
4	-0.0337	-0.0739	-0.3424	0.8447	96.9076	5.4877	5
5	0.3069	0.0594	0.1534	-0.1504	25.9957	5.2530	5
6	0.5935	-0.9492	-0.0568	0.3065	141.4174	4.7309	4
7	-0.0781	-0.1365	0.1985	-0.4500	41.8112	4.3468	4
8	0.5096	0.6965	0.8612	1.1598	58.1604	4.3457	4
9	1.0393	0.1189	0.1298	-0.3814	103.8019	4.1941	4
10	-0.8402	-0.5903	-0.5661	0.6175	185.3780	4.0894	4
11	0.2600	0.4863	0.2297	0.8905	40.5377	3.7026	3
12	0.3444	0.0147	0.5360	-1.1408	117.1287	3.6547	3
13	-0.5981	0.3141	-0.1056	-0.8512	126.5114	3.5880	3
14	-0.5203	-0.3588	-0.0351	0.5208	100.4585	3.2814	3
15	-0.8033	-0.0198	0.8381	0.7557	181.5376	3.2584	3
16	-0.4462	0.2337	0.3455	-0.1808	89.3629	3.2113	3

การหาคำตอบของฮิวริสติก

กระบวนการหาคำตอบของฮิวริสติกประกอบด้วย 7 องค์ประกอบหลัก ได้แก่ การสร้างคำตอบเริ่มต้น (Initial solutions), การทำลายคำตอบ (Destroy), การซ่อมแซมคำตอบ (Repair), การเสาะหาคำตอบ (Local search), การยอมรับคำตอบ (Acceptance criteria), การปรับค่าน้ำหนัก (Adaptive weight adjustment), และเงื่อนไขการหยุดหาคำตอบ (Stopping criteria) โดยมีแผนภาพแสดงขั้นตอนการทำงานดังแสดงในรูปที่ 3-8 และ Pseudocode ดังแสดงในรูปที่ 3-9



รูปที่ 3-8 แผนภาพแสดงขั้นตอนการทำงานของ ALNS

Pseudocode for ALNS

```

1: Function ALNS
2:    $sol_{best} = \{\emptyset_w, \emptyset_0\}$ 
3:    $sol \leftarrow sol_{best}$ 
4:    $R_{open} \leftarrow R$ 
5:    $R_{close} \leftarrow \emptyset$ 
6:   call INITIAL( $sol$ )
7:   repeat
8:     for  $seg = 1, \dots, nsegs$ 
9:       for  $iter = 1, \dots, niters$ 
10:        for  $i = 1, \dots, q$ 
11:          select and apply one destroy operator
12:          select and apply one repair operator
13:        end for
14:        if  $sol'$  is accepted by SA or  $f(sol) > f(sol_{best})$  then
15:           $sol \leftarrow sol'$ 
16:          if  $f(sol) > f(sol_{best})$  then
17:             $sol_{best} \leftarrow sol$ 
18:          end if
19:        end if
20:      end for
21:      update the weights of the ALNS and reset scores
22:      call LS( $sol$ )
23:      if  $sol'$  is accepted by SA or  $f(sol) > f(sol_{best})$  then
24:         $sol \leftarrow sol'$ 
25:        if  $f(sol) > f(sol_{best})$  then
26:           $sol_{best} \leftarrow sol$ 
27:        end if
28:      end if
29:    end for
30:  until stop criterion met
31: return  $sol_{best}$ 

```

รูปที่ 3-9 Pseudocode แสดงขั้นตอนการทำงานของ ALNS

กระบวนการหาคำตอบของฮิวริสติกเริ่มต้นด้วยการกำหนดคำตอบที่ดีที่สุด (sol_{best}) คือ การให้รถอยู่ที่ท่ารถตลอดวันทำการ ($sol_{best} = \emptyset_w, \emptyset_0$) เมื่อ \emptyset_w คือ ตำแหน่งท่ารถที่รอบรถ w ซึ่งมีข้อกำหนดให้ปล่อยรถที่เวลา w เท่านั้น จากนั้นจึงอัปเดตคำตอบสำหรับการค้นหา (sol) ด้วย sol_{best} เพื่อสร้างคำตอบเริ่มต้น (Initial solution) และนำไปเข้ากระบวนการค้นหาคำตอบในลำดับถัดไป เซ็ตของคำสั่งของลูกค้าที่ยังไม่ถูกมอบหมาย (R_{open}) ถูกกำหนดให้เท่ากับเซ็ตคำสั่งทั้งหมดในข้อมูลนำเข้าของปัญหา (R) ส่วนเซ็ตของคำสั่งของลูกค้าที่ถูกมอบหมายแล้ว (R_{close}) ถูกกำหนดให้เป็นเซ็ตว่าง ดังแสดงไว้ในบรรทัดที่ 2 ถึงบรรทัดที่ 6 ของรูปที่ 3-9

การค้นหาคำตอบจะถูกแบ่งออกเป็น Segment จำนวน $nsegs$ โดยแต่ละ Segment ประกอบด้วยรอบการหาคำตอบ (Iterations) จำนวน $niters$ ที่จะทำลายคำตอบ (Destroy) และซ่อมแซมคำตอบ (Repair) จำนวน q คำสั่งอย่าง ต่อเนื่อง ดังแสดงไว้ในบรรทัดที่ 8 ถึงบรรทัดที่ 13 ของรูปที่ 3-9

เมื่อการค้นหาแต่ละ Segment สิ้นสุดลง คำนวณน้ำหนักที่ใช้ในการกำหนด ความน่าจะเป็นของการเลือกใช้วิธีทำลายคำตอบ และวิธีซ่อมแซมคำตอบจะถูก ปรับเปลี่ยนด้วยคะแนนสะสม และจำนวนครั้งที่วิธีดังกล่าวถูกใช้ใน Segment ที่ ผ่านมา แล้วคำตอบจะผ่านกระบวนการปรับปรุงคำตอบ (Local search: LS) เพื่อ ปรับปรุงคำตอบให้ดีขึ้น ดังแสดงไว้ในบรรทัดที่ 21 และบรรทัดที่ 22 ของรูปที่ 3-9 ตามลำดับ

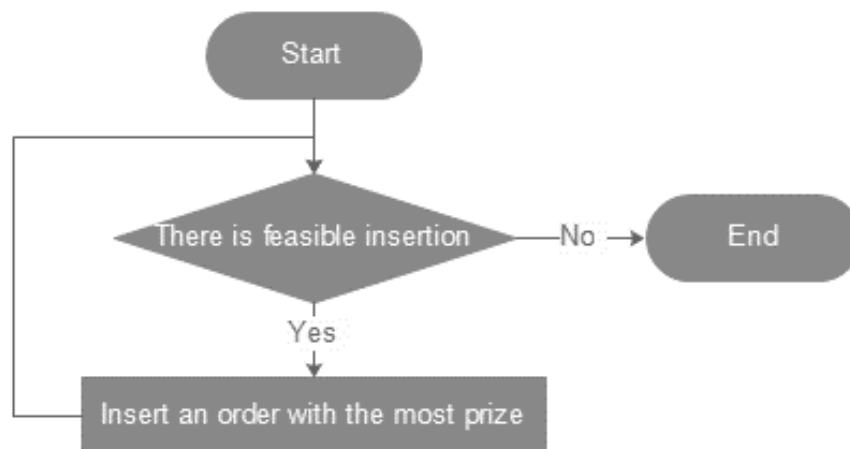
คำตอบใหม่ (sol') ที่เกิดขึ้นในแต่ละรอบการหาคำตอบจะถูกยอมรับก็ ต่อเมื่อค่า Objective ของคำตอบใหม่ ($f(sol')$) มีค่ามากกว่าค่า Objective ของ คำตอบที่ดีที่สุด ($f(sol_{best})$) หรือเมื่อคำตอบใหม่ถูกยอมรับด้วย Simulated Annealing (SA) จากนั้นจึงอัปเดต sol_{best} ด้วย sol' ถ้า $f(sol')$ มีค่ามากกว่า $f(sol_{best})$ ดังแสดงไว้ในบรรทัดที่ 14 ถึงบรรทัดที่ 19 และบรรทัดที่ 23 ถึงบรรทัด ที่ 28 ของรูปที่ 3-9

สุดท้ายนี้ การหาคำตอบของ ALNS จะสิ้นสุดลงเมื่อเป็นไปตามเงื่อนไขการ หยุดหาคำตอบที่กำหนดไว้ (Stop criterion) เพื่อประหยัดเวลาในการหาคำตอบ ดัง แสดงไว้ในบรรทัดที่ 30 ของรูปที่ 3-9

จุฬาลงกรณ์มหาวิทยาลัย

การสร้างคำตอบเริ่มต้น (Initial Solution)

การสร้างคำตอบเริ่มต้นเริ่มต้นด้วยการเลือกคำสั่ง r จาก R_{open} ที่มีค่าตอบแทนสูง ที่สุด จากนั้นจึงแทรกตำแหน่งท่ารถที่มีรอบรถตรงกับรอบการปรากฏของคำสั่ง r (\mathcal{O}_{r_i}) ลงใน sol เพื่อทำหน้าที่เป็นจุดเริ่มต้นสำหรับการปล่อยรถเพื่อให้บริการคำสั่ง r โดยคำสั่ง r จะถูก แทรก ในตำแหน่งที่เกิดค่าเดินทางน้อยที่สุดของ sol และเป็นไปตามข้อกำหนดของปัญหา สุดท้ายจึงอัปเดต R_{open} กับ R_{close} ทั้งนี้การเลือกคำสั่ง และการแทรกคำสั่ง จะเกิดขึ้นอย่าง ซ้ำ ๆ จนไม่สามารถแทรกคำสั่งใด ๆ ใน sol ได้อีก โดยมีแผนภาพแสดงขั้นตอนการทำงานดัง แสดงในรูปที่ 3-10 และ Pseudocode ดังแสดงในรูปที่ 3-11



รูปที่ 3-10 แผนภาพแสดงขั้นตอนการสร้างคำตอบเริ่มต้น



Pseudocode for Initial Solution Construction

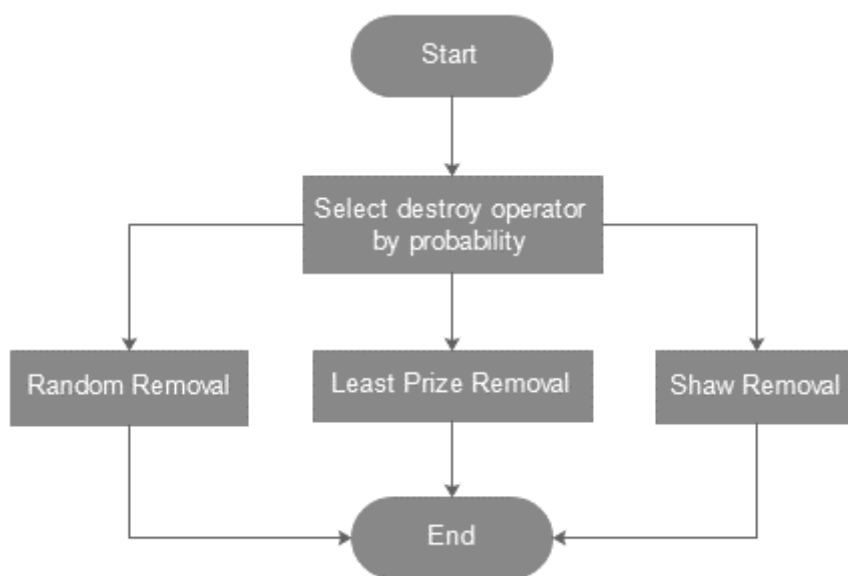
```

1: Function INITIAL(sol)
2:   LIST = set of sorted  $r \in R_{open}$  in decreasing order of prizes
3:   repeat
4:      $r = LIST(1)$ 
5:     if  $\emptyset_{\tau_r} \notin R_{close}$  then
6:       insert  $\emptyset_{\tau_r}$  in a feasible position of sol
7:     end if
8:     insert  $r$  in a feasible position of sol with the least insertion cost
9:      $LIST \leftarrow LIST \setminus \{r\}$ 
10:     $R_{open} \leftarrow R_{open} \setminus \{r\}$ 
11:     $R_{close} \leftarrow R_{close} \cup \{r\}$ 
12:  until no more feasible insertion or LIST is empty
13: return sol
  
```

รูปที่ 3-11 Pseudocode แสดงขั้นตอนการสร้างคำตอบเริ่มต้น

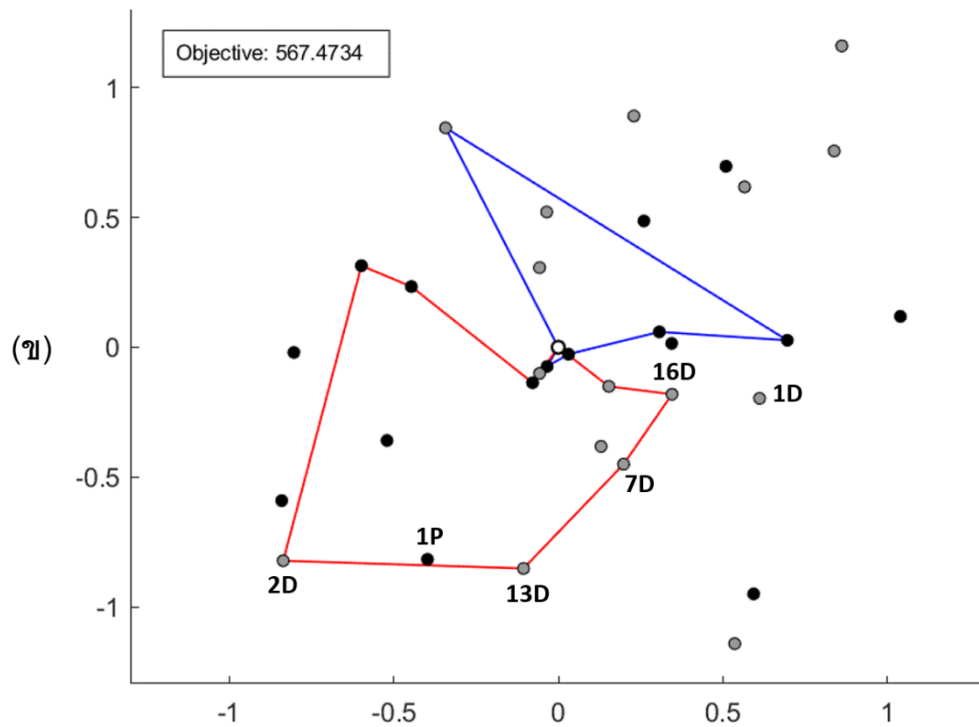
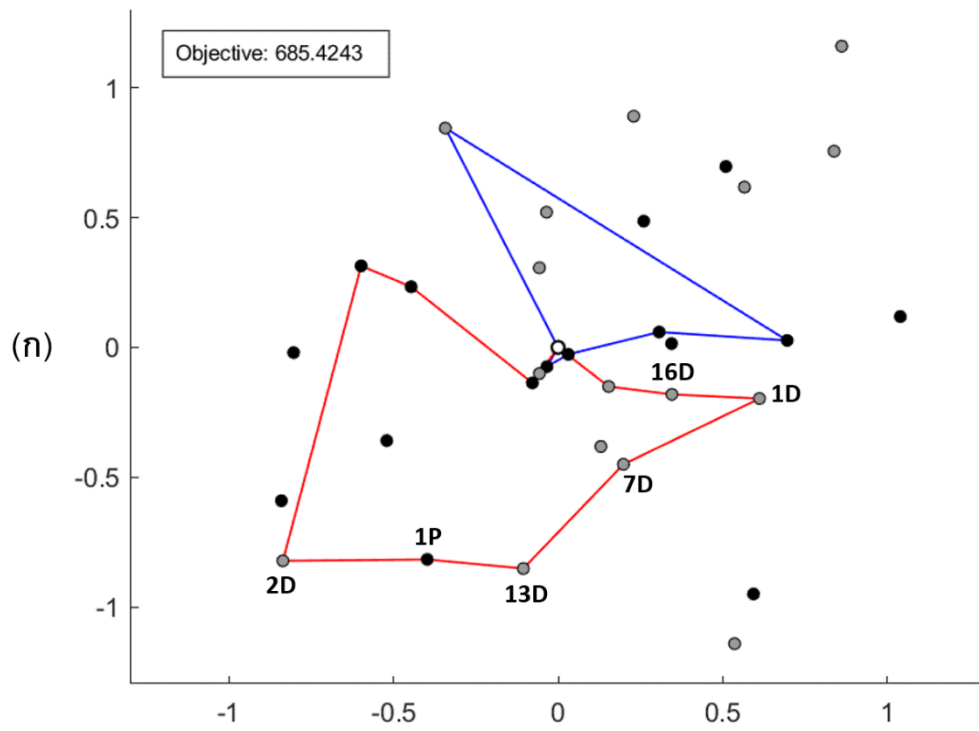
การทำลายคำตอบ (Destroy)

การทำลายคำตอบ คือ การเปลี่ยนแปลงคำตอบปัจจุบันโดยนำคำสั่งของลูกค้าที่อยู่ใน R_{close} ออกจาก sol จำนวน q คำสั่ง ตามนโยบายที่กำหนดขึ้น ซึ่งในงานวิจัยนี้ใช้นโยบายการทำลายคำตอบทั้งหมด 3 แบบ ได้แก่ Random removal, Least prize removal, และ Shaw removal โดยมีแผนภาพแสดงขั้นตอนการทำงานดังแสดงในรูปที่ 3-12



รูปที่ 3-12 แผนภาพแสดงขั้นตอนการทำลายคำตอบ

การทำลายคำตอบจะนำทั้งตำแหน่งรับ และตำแหน่งส่งพัสดุของคำสั่งที่ถูกเลือกออกจาก sol โดยเส้นทางใหม่จะมุ่งไปหาตำแหน่งถัดไปใน sol แทน ซึ่งเส้นทางใหม่นี้จะมีระยะทางน้อยกว่าเส้นทางเดิมเสมอ จากนั้นจึงอัปเดต R_{open} กับ R_{close} ด้วยคำสั่งที่ถูกนำออกจากคำตอบ ทั้งนี้ตัวอย่างการทำลายคำสั่งลำดับที่ 1 สามารถแสดงได้ด้วยรูปที่ 3-13 เมื่อ 1P และ 1D คือตำแหน่งรับ และตำแหน่งส่งพัสดุของคำสั่งลำดับที่ 1 ตามลำดับ



รูปที่ 3-13 ตัวอย่างการทำลายคำตอบ
(ก) คำตอบก่อนถูกทำลาย และ (ข) คำตอบหลังถูกทำลาย

○ Random removal

Random removal เป็นวิธีการทำลายคำตอบโดยนำคำสั่งของลูกค้าออกจาก sol ด้วยวิธีการสุ่ม โดยมีขั้นตอนดังแสดงในรูปที่ 3-14

Pseudocode for Random Removal

```

1: Function RANDOM( $sol$ )
2:   repeat
3:     remove a random request  $r \in R_{close}$  from  $sol$ 
4:      $R_{open} \leftarrow R_{open} \cup \{r\}$ 
5:      $R_{close} \leftarrow R_{close} \setminus \{r\}$ 
6:   until one request is removed or no more possible removal
7: return  $sol'$ 

```

รูปที่ 3-14 Pseudocode แสดงขั้นตอนการทำงานของ Random removal

○ Least prize removal

Least prize removal เป็นวิธีการทำลายคำตอบโดยนำคำสั่งของลูกค้าที่มีค่าตอบแทนต่ำที่สุดออกจาก sol โดยมีขั้นตอนดังแสดงในรูปที่ 3-15

Pseudocode for Least Prize Removal

```

1: Function LEASTPRIZE( $sol$ )
2:   repeat
3:      $LIST =$  set of sorted unserved requests in  $sol$  in increasing order of prizes
4:     remove  $LIST(1)$  from  $sol$ 
5:      $R_{open} \leftarrow R_{open} \cup LIST(1)$ 
6:      $R_{close} \leftarrow R_{close} \setminus LIST(1)$ 
7:   until one request is removed or no more possible removal
8: return  $sol'$ 

```

รูปที่ 3-15 Pseudocode แสดงขั้นตอนการทำงานของ Least prize removal

○ Shaw removal

Shaw removal เป็นวิธีการทำลายคำตอบโดยนำคำสั่งของลูกค้า 2 คำสั่ง ที่ส่งผลต่อค่า Objective ใกล้เคียงกันออกจาก sol โดยค่าความคล้ายคลึงของคำสั่ง i และคำสั่ง j (r_{ij}^*) คำนวณได้จากผลต่างสัมบูรณ์ของค่า Objective ของ sol เมื่อเพิ่มคำสั่ง i ลงในตำแหน่งที่ใช้ค่าแทรกงานต่ำที่สุด ($f(sol_i)$) กับค่า Objective ของ sol เมื่อเพิ่มคำสั่ง j ลงในตำแหน่งที่ใช้ค่าแทรกงานต่ำที่สุด ($f(sol_j)$) ตามสมการที่ (3.3) ทั้งนี้ขั้นตอนการทำงานของ Shaw removal สามารถแสดงได้ด้วยรูปที่ 3-16

$$r_{ij}^* = |f(sol_i) - f(sol_j)| \quad (3.3)$$

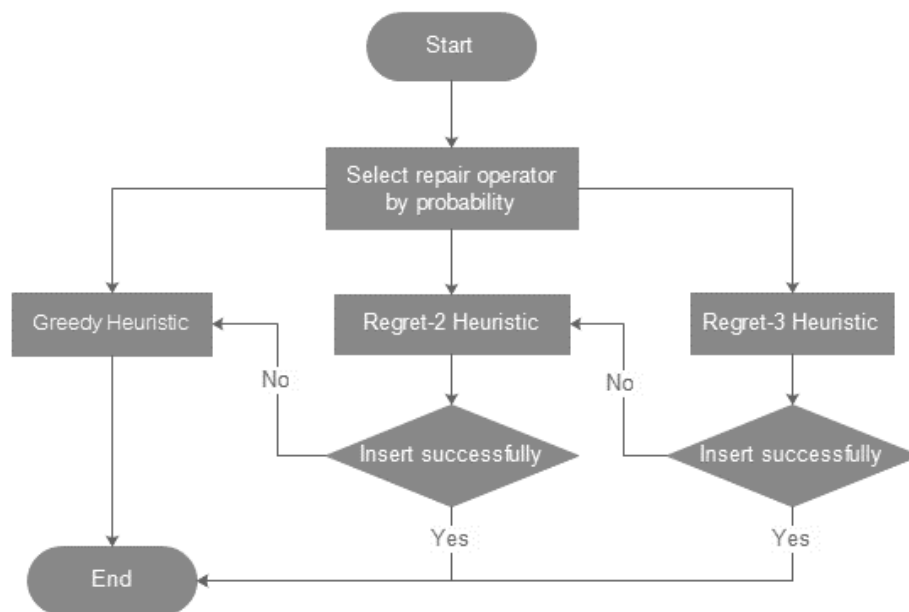
Pseudocode for Shaw Removal

- 1: **Function** SHAWREMOVAL(*sol*)
 - 2: calculate r_{ij}^* for all pair of request in R_{close}
 - 3: **repeat**
 - 4: remove a pair of request with the lowest r_{ij}^* from *sol*
 - 5: update R_{open} and R_{close}
 - 6: **until** two requests are removed or no more possible removal
 - 7: **return** *sol'*
-

รูปที่ 3-16 Pseudocode แสดงขั้นตอนการทำงานของ Shaw removal

การซ่อมแซมคำตอบ (Repair)

การซ่อมแซมคำตอบ คือ การนำคำสั่งของลูกค้าจาก R_{open} มาแทรกใน *sol* ตามนโยบายที่กำหนดขึ้น เพื่อปรับปรุงค่า Objective ซึ่งในงานวิจัยนี้ใช้วิธีซ่อมแซมคำตอบทั้งสิ้น 3 แบบ ได้แก่ Greedy insertion, Regret-2 insertion และ Regret-3 insertion โดยมีแผนภาพแสดงขั้นตอนการทำงานดังแสดงในรูปที่ 3-17



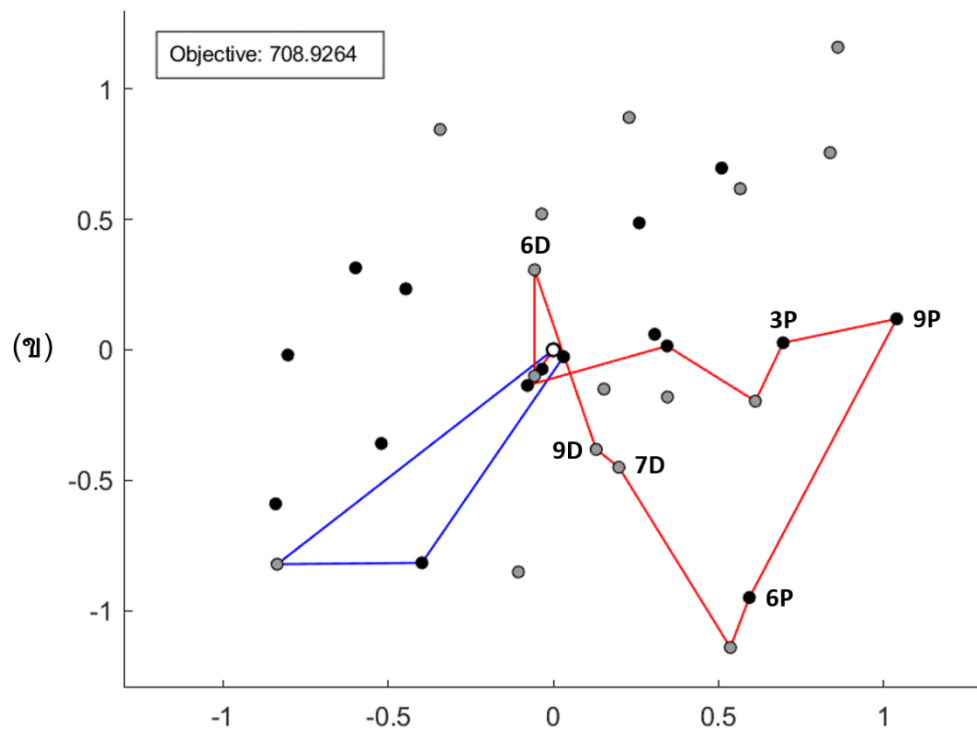
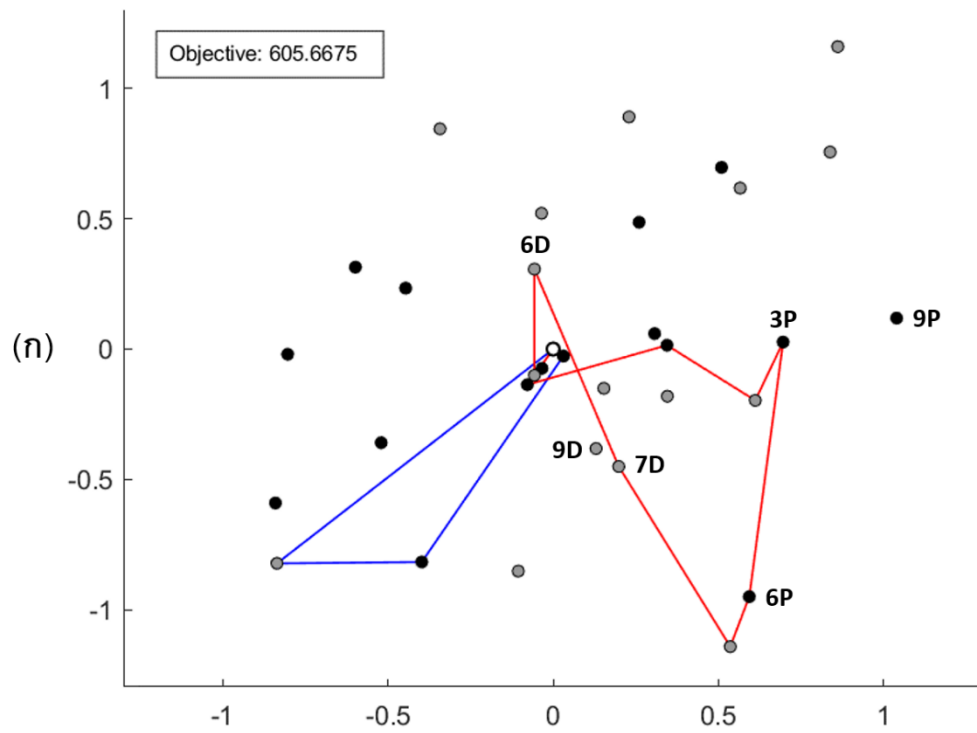
รูปที่ 3-17 แผนภาพแสดงขั้นตอนการซ่อมแซมคำตอบ

การซ่อมแซมคำตอบเริ่มต้นจากการนำตำแหน่งท่ารถที่ไม่มีการปล่อยรถออกจาก sol เพื่อให้สามารถแทรกคำสั่งอื่นลงในคำตอบได้ โดยไม่ขัดกับกรอบเวลาของตำแหน่งท่ารถเหล่านั้น จากนั้นจึงแทรกตำแหน่งท่ารถที่มีรอบรถตรงกับรอบการปรากฏของคำสั่ง r (Φ_{r_r}) ลงใน sol เพื่อให้มีจุดเริ่มต้นปล่อยรถสำหรับคำสั่ง r สุดท้ายจึงอัปเดต R_{open} กับ R_{close} ด้วยคำสั่งที่นำมาซ่อมแซมคำตอบ

สำหรับการปรับปรุงของค่า Objective จากการแทรกคำสั่ง r เกิดจากผลต่างของคำตอบแทนของคำสั่งที่ r กับค่าแทรกงานของตำแหน่งรับพัสดุของคำสั่ง r (p_r) และตำแหน่งส่งพัสดุของคำสั่ง r (d_r) ใน sol โดยคำนวณได้จากสมการที่ (3.4) เมื่อ P_r คือคำตอบแทนของคำสั่ง r , i คือตำแหน่งที่อยู่ก่อนหน้าตำแหน่ง p_r , j คือตำแหน่งที่อยู่ถัดจากตำแหน่ง p_r , m คือตำแหน่งที่อยู่ก่อนหน้าตำแหน่ง d_r , n คือตำแหน่งที่อยู่ถัดจากตำแหน่ง d_r , และ $|a_{ab}|$ คือ ระยะห่างระหว่างตำแหน่ง a และตำแหน่ง b

$$f(sol') - f(sol) = P_r + |a_{ip_r}| + |a_{p_rj}| + |a_{md_r}| + |a_{d_rn}| - |a_{ij}| - |a_{mn}| \quad (3.4)$$

ตัวอย่างการซ่อมแซมคำตอบด้วยคำสั่งลำดับที่ 8 สามารถแสดงได้ด้วยรูปที่ 3-18 โดยการแทรกคำสั่งลำดับที่ 8 ทำให้ค่า Objective ของคำตอบเพิ่มขึ้น 103.2589 หน่วย



รูปที่ 3-18 ตัวอย่างการซ่อมแซมคำตอบ

(ก) คำตอบก่อนถูกซ่อมแซม และ (ข) คำตอบหลังถูกซ่อมแซม

○ Greedy insertion

Greedy insertion เป็นวิธีการซ่อมแซมคำตอบด้วยการนำคำสั่งของลูกค้าจาก R_{open} ที่จะทำให้ $f(sol)$ เพิ่มมากที่สุด แทรกใน sol (Avci and Avci, 2019) โดยขั้นตอนการทำงานของ Greedy insertion สามารถแสดงได้ด้วยรูปที่ 3-19

Pseudocode for Greedy Insertion Heuristic	
1:	Function <i>GREEDY</i> (<i>sol</i>)
2:	remove all unnecessary \emptyset from <i>sol</i>
3:	repeat
4:	select request $r \in R_{open}$
5:	if $\emptyset_{r_r} \notin R_{close}$ then
6:	insert \emptyset_{r_r} in a feasible position of <i>sol</i>
7:	end if
8:	insert request r in a feasible position of <i>sol</i> where $f(sol)$ improves the most
9:	$R_{open} \leftarrow R_{open} \setminus \{r\}$
10:	$R_{close} \leftarrow R_{close} \cup \{r\}$
11:	until no more feasible insertion or R_{open} is empty
12:	return <i>sol'</i>

รูปที่ 3-19 Pseudocode แสดงขั้นตอนการทำงานของ Greedy insertion

○ Regret-2 insertion

Regret-2 insertion เป็นวิธีการซ่อมแซมคำตอบด้วยการนำคำสั่งจาก R_{open} ที่มีค่า Regret-2 (c_{i2}^*) สูงที่สุด ที่จะทำให้ค่า $f(sol)$ เกิดการปรับปรุงมากที่สุดเป็นอันดับสอง แทรกใน sol ซึ่ง c_{i2}^* มีค่าเท่ากับผลต่างของค่า Objective เมื่อแทรกคำสั่ง i ในตำแหน่งที่มีค่าแทรกงานน้อยที่สุดของรอบรถอันดับสอง ($\Delta f_{i,x2}$) กับค่า Objective เมื่อแทรกคำสั่ง i ในตำแหน่งที่มีค่าแทรกงานน้อยที่สุดของรอบรถอันดับหนึ่ง ($\Delta f_{i,x1}$) (Avci and Avci, 2019) ดังแสดงในสมการที่ (3.5) เมื่อ $x_{ik} \in \{1, \dots, \omega\}$ ระบุรอบรถลำดับที่ k ที่คำสั่ง i มีค่าแทรกงานน้อยที่สุด จากนั้นจึงอัปเดต R_{open} กับ R_{close} ด้วยคำสั่งที่นำมาซ่อมแซมคำตอบ โดยขั้นตอนการทำงานของ Regret-2 insertion สามารถแสดงได้ด้วยรูปที่ 3-20

$$c_{i2}^* = \Delta f_{i,x2} - \Delta f_{i,x1} \quad (3.5)$$

Pseudocode for Regret – 2 Insertion Heuristic

```

1: Function REGRET2(sol)
2:   remove all unnecessary  $\emptyset$  from sol
3:   calculate  $c_{i2}^*$  for all request  $r \in R_{open}$ 
4:   repeat
5:     select request  $r \in R_{open}$  with the most  $c_{i2}^*$ 
6:     if  $\emptyset_{r_r} \notin R_{close}$  then
7:       insert  $\emptyset_{r_r}$  in a feasible position of sol
8:     end if
9:     insert request  $r$  in a feasible position of sol with the least insertion cost
10:     $R_{open} \leftarrow R_{open} \setminus \{r\}$ 
11:     $R_{close} \leftarrow R_{close} \cup \{r\}$ 
12:  until no more feasible insertion or  $R_{open}$  is empty
13: return sol'

```

รูปที่ 3-20 Pseudocode แสดงขั้นตอนการทำงานของ Regret-2 insertion

○ Regret-3 insertion

Regret-3 insertion เป็นวิธีการซ่อมแซมคำตอบด้วยการนำคำสั่งจาก R_{open} ที่มีค่า Regret-3 (c_{i3}^*) สูงที่สุด ที่จะทำให้ค่า $f(sol)$ เกิดการปรับปรุงมากที่สุดเป็นอันดับสาม แทรกใน sol ซึ่ง c_{i3}^* มีค่าเท่ากับผลต่างของค่า Objective เมื่อแทรกคำสั่ง i ในตำแหน่งที่มีค่าแทรกงานน้อยที่สุดของรอบรถอันดับสาม ($\Delta f_{i,x3}$) กับค่า Objective เมื่อแทรกคำสั่ง i ในตำแหน่งที่มีค่าแทรกงานน้อยที่สุดของรอบรถอันดับหนึ่ง ($\Delta f_{i,x1}$) (Avci and Avci, 2019) ดังแสดงในสมการที่ (3.6) เมื่อ $x_{ik} \in \{1, \dots, \omega\}$ ระบุรอบรถลำดับที่ k ที่คำสั่ง i มีค่าแทรกงานน้อยที่สุด จากนั้นจึงอัปเดต R_{open} กับ R_{close} ด้วยคำสั่งที่นำมาซ่อมแซมคำตอบ โดยขั้นตอนการทำงานของ Regret-3 insertion สามารถแสดงได้ด้วยรูปที่ 3-21

$$c_{i3}^* = \Delta f_{i,x1} - \Delta f_{i,x3} \quad (3.6)$$

Pseudocode for Regret – 3 Insertion Heuristic

```

1: Function REGRET3(sol)
2:   remove all unnecessary  $\emptyset$  from sol
3:   calculate  $c_{i3}^*$  for all request  $r \in R_{open}$ 
4:   repeat
5:     select request  $r \in R_{open}$  with the most  $c_{i3}^*$ 
6:     if  $\emptyset_{r_r} \notin R_{close}$  then
7:       insert  $\emptyset_{r_r}$  in a feasible position of sol
8:     end if
9:     insert request  $r$  in a feasible position of sol with the least insertion cost
10:     $R_{open} \leftarrow R_{open} \setminus \{r\}$ 
11:     $R_{close} \leftarrow R_{close} \cup \{r\}$ 
12:  until no more feasible insertion or  $R_{open}$  is empty
13: return sol'

```

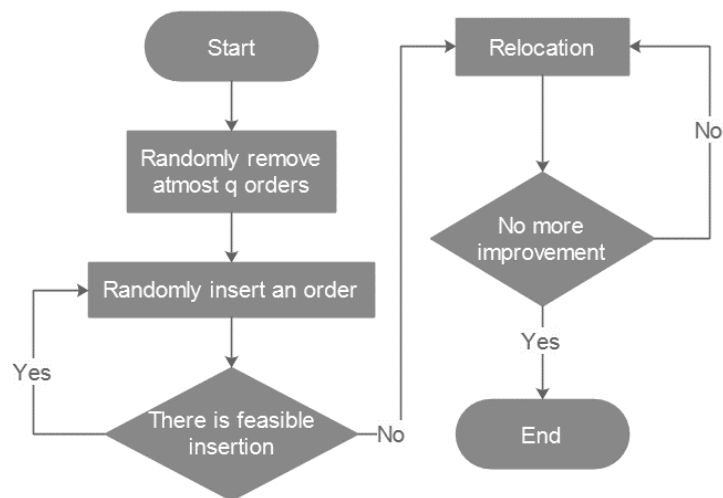
รูปที่ 3-21 Pseudocode แสดงขั้นตอนการทำงานของ Regret-3 insertion

การเสาะหาคำตอบ (Local search)

การเสาะหาคำตอบ คือ การค้นหาคำตอบที่อยู่ใกล้เคียงกับคำตอบปัจจุบันที่มีการพัฒนาในทางที่ดีขึ้นของค่า Objective โดยฮิวริสติกปรับปรุงคำตอบอย่างง่าย ซึ่งในงานวิจัยเลือกใช้ คือ การนำคำสั่งออกจากคำตอบ (Request removal), การเพิ่มคำสั่งในคำตอบ (Request insertion), และการเปลี่ยนลำดับงาน (Relocation) ในการปรับปรุงคำตอบ

การปรับปรุงคำตอบจะเริ่มจาก การนำคำสั่งของลูกค้าออกจาก sol เป็นจำนวน $n^* \in \{1, 2, \dots, q\}$ คำสั่ง ด้วยวิธีการสุ่ม จากนั้นจึงสุ่มคำสั่ง $r \in R_{open}$ เพื่อนำมาเพิ่มใน sol ณ ตำแหน่งที่เกิดค่าเดินทางน้อยที่สุด และเป็นไปตามข้อกำหนดของปัญหา จนไม่สามารถเพิ่มคำสั่งใดได้อีก แล้วจึงทำการค้นหาคำตอบที่มีค่าเดินทางน้อยลงด้วยการเปลี่ยนลำดับงาน

ทั้งนี้การเสาะหาคำตอบจะเกิดขึ้นซ้ำไปมา จนคำตอบไม่เกิดการปรับปรุงเป็นจำนวนรอบที่กำหนดไว้ จึงสิ้นสุดการปรับปรุงคำตอบ โดยมีแผนภาพแสดงขั้นตอนการทำงานดังแสดงในรูปที่ 3-22 และ Pseudocode ดังแสดงในรูปที่ 3-23



รูปที่ 3-22 แผนภาพแสดงขั้นตอนการทำงานของ Local search



Pseudocode for Local Search Algorithm

```

1: Function  $LS(sol)$ 
2:   repeat
3:     randomly remove  $n^* \in \{1,2, \dots, q\}$  request(s) from  $sol$ 
4:     repeat
5:       remove all unnecessary  $\emptyset$  from  $sol$ 
6:       select random request  $r \in R_{open}$ 
7:       if  $\emptyset_{\tau_r} \in R_{close}$  then
8:         insert  $\emptyset_{\tau_r}$  in feasible position
9:       end if
10:      insert request  $r$  into  $sol$  with the least insertion cost
11:       $R_{open} \leftarrow R_{open} \setminus \{r\}$ 
12:    until no more feasible insertion or  $R_{open}$  is empty
13:    repeat
14:      call  $RELOCATION(sol)$ 
15:    until no more improvement for a set amount of iterations
16:  until no more improvement for a set amount of iterations
17: return  $sol'$ 
  
```

รูปที่ 3-23 Pseudocode แสดงขั้นตอนการทำงานของ Local search

○ การเปลี่ยนลำดับงาน (Relocation)

การเปลี่ยนลำดับงาน คือ การค้นหาคำตอบที่อยู่ใกล้เคียงคำตอบปัจจุบันที่มีการพัฒนาในทางที่ดีขึ้นของค่า Objective ด้วยการสลับตำแหน่ง i ใน sol แล้วย้ายตำแหน่งงานไปยังตำแหน่งใหม่ที่เกิดค่าเดินทางน้อยที่สุด และเป็นไปตามข้อกำหนดของปัญหา โดยขั้นตอนการทำงานของ Relocation สามารถแสดงได้ด้วยรูปที่ 3-24

Pseudocode for Relocation Algorithm

```

1: Function RELOCATION(sol)
2:   repeat
3:     randomly remove node  $i \in sol$ 
4:     insert node  $i$  into  $sol$  with the least insertion cost
5:   until no more improvement for a set amount of iterations
6: return  $sol'$ 

```

รูปที่ 3-24 Pseudocode แสดงขั้นตอนการทำงานของ Relocation

การยอมรับคำตอบ (Acceptance criteria)

การยอมรับคำตอบที่มีค่า Objective ต่ำกว่าคำตอบก่อนหน้า (Degrading solution) ด้วยความน่าจะเป็น มีความสำคัญใน ALNS เพราะจะช่วยให้เกิดการค้นหาคำตอบที่หลากหลายมากขึ้น ซึ่งงานวิจัยนี้เลือกใช้ SA ในการคำนวณความน่าจะเป็นในการยอมรับคำตอบ (p_{sol}) ตามสมการที่ (3.7) เมื่อ $f(sol^*)$ คือ Objective ของคำตอบที่ถูกยอมรับล่าสุด, $f(sol)$ คือ Objective ของคำตอบปัจจุบัน, และ $T > 0$ คือ อุณหภูมิ (Temperature) ที่จะลดลงในทุกรอบที่มีการยอมรับคำตอบด้วย SA โดยมีความสัมพันธ์ตามสมการที่ (3.8) เมื่อ c คือ อัตราการเย็นตัว (Cooling rate) ที่มีค่าอยู่ระหว่างศูนย์กับหนึ่ง

$$p_{sol} = e^{-\frac{f(sol^*) - f(sol)}{T}} \quad (3.7)$$

$$T' = T \cdot c \quad (3.8)$$

โดยในงานวิจัยนี้อุณหภูมิเริ่มต้น (T_{start}) ที่ใช้ใน Segment แรกของ ALNS จะถูกคำนวณขึ้นจากความน่าจะเป็นเท่ากับ 0.5 ในการยอมรับคำตอบที่มีค่า Objective ต่ำกว่าคำตอบเริ่มต้นเท่ากับ 50 เปอร์เซ็นต์

การปรับค่าน้ำหนัก (Adaptive weight adjustment)

ค่าน้ำหนักของแต่ละวิธีทำลายคำตอบ และวิธีซ่อมแซมคำตอบจะถูกปรับในตอนท้ายของแต่ละ Segment ด้วยคะแนนสะสม และจำนวนครั้งที่วิธีดังกล่าวถูกเลือกใช้ ใน Segment ที่ผ่านมา เพื่อให้ ALNS เลือกวิธีที่ปรับปรุงค่า Objective ได้อย่างมีประสิทธิภาพมากขึ้นใน Segment ถัดไป

หากมีวิธีการทำลายคำตอบ หรือวิธีการซ่อมแซมคำตอบจำนวน k วิธี แต่ละวิธีจะมีค่าน้ำหนัก $w_i, i \in \{1, 2, \dots, k\}$ โดยที่ความน่าจะเป็นในการเลือกวิธี j สามารถคำนวณได้จากสมการที่ (3.9)

$$p_j = \frac{w_j}{\sum_{i=1}^k w_i} \quad (3.9)$$

ค่าน้ำหนักของแต่ละวิธีใน Segment แรกของ ALNS ถูกกำหนดให้มีค่าเท่ากับ 1 แต่ใน Segment $j > 1$ จะมีการปรับค่าน้ำหนักตามสมการที่ (3.10) เมื่อ π_i คือ คะแนนทั้งหมดของวิธี i ที่ได้รับใน Segment ที่ผ่านมา, และ θ_i คือ จำนวนครั้งทั้งหมดที่วิธี i ถูกใช้ใน Segment ที่ผ่านมา, และ $r \in [0, 1]$ คือ อัตราส่วนในการปรับค่าน้ำหนัก (Reaction factor) ซึ่งถ้า r ถูกกำหนดให้มีค่าเป็นศูนย์ แสดงว่า ALNS ไม่มีการปรับค่าน้ำหนักระหว่างกระบวนการหาคำตอบ

$$w_{i,j+1} = w_{i,j}(1 - r) + r \frac{\pi_i}{\theta_i} \quad (3.10)$$

คะแนนเริ่มต้นของแต่ละวิธีจะถูกกำหนดให้มีค่าเท่ากับศูนย์ในทุกการเริ่มต้นของ Segment จากนั้นคะแนนจะเพิ่มขึ้นด้วย $\sigma_1, \sigma_2, \sigma_3$, หรือ σ_4 ตามเงื่อนไขที่กำหนดในตารางที่ 3-4 โดยคะแนนของวิธีทำลายคำตอบ และวิธีซ่อมแซมคำตอบจะถูกเพิ่มในแต่ละ Iteration ไปพร้อมกัน

ตารางที่ 3-4 พารามิเตอร์สำหรับให้คะแนน Operator ใน ALNS

พารามิเตอร์	เงื่อนไข
σ_1	รอบการหาคำตอบล่าสุดมีการยอมรับคำตอบที่เป็น Global best
σ_2	รอบการหาคำตอบล่าสุดมีการยอมรับคำตอบที่ดีกว่าคำตอบในปัจจุบัน และเป็นคำตอบที่ไม่เคยพบมาก่อน
σ_3	รอบการหาคำตอบล่าสุดมีการยอมรับคำตอบที่ดีกว่าคำตอบในปัจจุบัน แต่เป็นคำตอบที่เคยพบมาก่อน
σ_4	รอบการหาคำตอบล่าสุดมีการยอมรับคำตอบที่แย่กว่าคำตอบในปัจจุบัน

เงื่อนไขการหยุดหาคำตอบ (Stopping criteria)

ALNS จะหยุดหาคำตอบเมื่อไม่สามารถหาคำตอบที่มีคุณภาพสูงกว่าคำตอบปัจจุบันเป็นจำนวน Iteration ที่กำหนดไว้เพื่อประหยัดเวลาการหาคำตอบ (Fail count limit) หรือหยุดหาคำตอบเมื่อ ALNS ใช้เวลาหาคำตอบเกินเวลาที่กำหนด (Run time limit)

3.3.5 ตัวอย่างการหาคำตอบของฮิวริสติก

พิจารณาข้อมูลนำเข้าสำหรับตัวอย่างการหาคำตอบของฮิวริสติกที่มีจำนวนคำสั่งทั้งหมดที่เป็นไปได้เท่ากับ 16 คำสั่ง, 6 รอบรถ, และระยะเวลาทำการ 10 ชั่วโมง ดังแสดงในตารางที่ 3-5 โดยตำแหน่งของท่ารถอยู่ที่จุดกำเนิด (0,0) และตำแหน่งของแต่ละคำสั่งถูกสุ่มในพื้นที่รอบนอกของท่ารถที่ใช้เวลาเดินทางไป-กลับไม่เกิน 2.5 ชั่วโมง เวลาการปรากฏของแต่ละคำสั่งถูกสุ่มในครั้งแรกของระยะเวลาทำการ (ระหว่างรอบรถที่ 3 และ 5) นอกจากนี้ค่า α ถูกกำหนดให้เท่ากับ 100 เพื่อให้มีระดับการให้บริการ (Service level) สูงที่สุด

ตารางที่ 3-5 ข้อมูลนำเข้าของตัวอย่างการหาคำตอบด้วยฮิวริสติก

Order No.	Pickup node		Delivery node		Order prize	Arrival time	Arrival wave
	X	Y	X	Y			
1	-0.3975	-0.8164	0.6115	-0.1968	118.3966	5.9546	5
2	0.0310	-0.0265	-0.8356	-0.8216	117.6104	5.8591	5
3	0.6958	0.0269	-0.0566	-0.1004	76.3094	5.7914	5
4	-0.0337	-0.0739	-0.3424	0.8447	96.9076	5.4877	5
5	0.3069	-0.0594	0.1534	-0.1504	25.9957	5.2530	5
6	0.5935	-0.9492	-0.0568	0.3065	141.4174	4.7309	4
7	-0.0781	-0.1365	0.1985	-0.4500	41.8112	4.3468	4
8	0.5096	0.6965	0.8612	1.1598	58.1604	4.3457	4
9	1.0393	0.1189	0.1298	-0.3814	103.8019	4.1941	4
10	-0.8402	-0.5903	0.5661	0.6175	185.3780	4.0894	4
11	0.2600	0.4863	0.2297	0.8905	40.5377	3.7026	3
12	0.3444	0.0147	0.5360	-1.1408	117.1287	3.6547	3
13	-0.5981	0.3141	-0.1056	-0.8512	126.5114	3.5880	3
14	-0.5203	-0.3588	-0.0351	0.5208	100.4585	3.2814	3
15	-0.8033	-0.0198	0.8381	0.7557	181.5376	3.2584	3
16	-0.4462	0.2337	0.3455	-0.1808	89.3629	3.2113	3

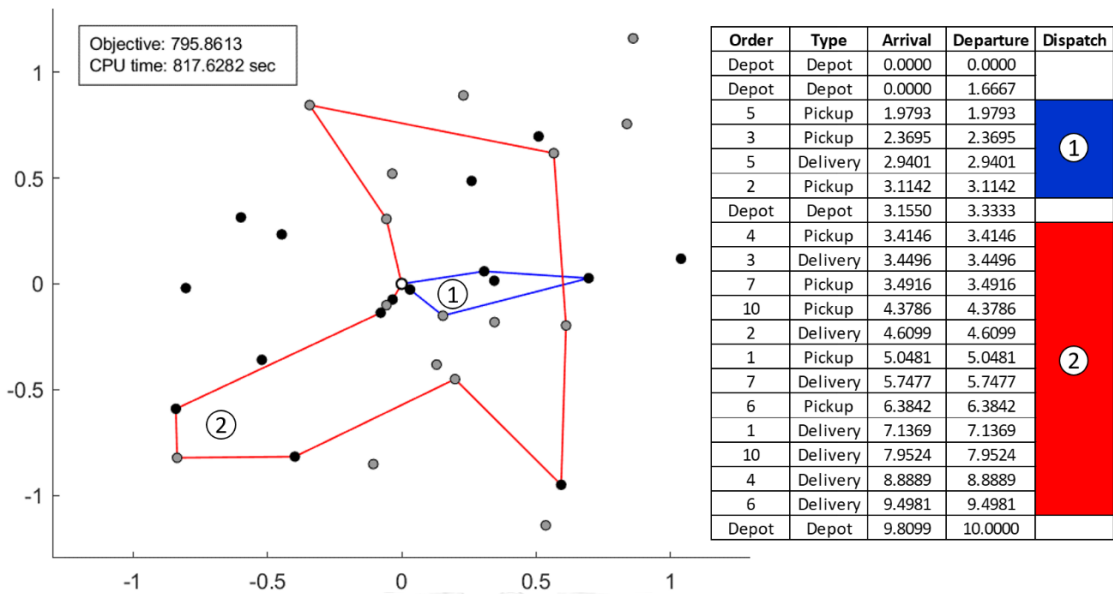
สำหรับ ALNS ถูกตั้งค่าให้มีจำนวนรอบการค้นหาค่าตอบสูงสุดจำนวน 5000 รอบ และมีจำนวน Segment ทั้งหมด 50 Segment โดยแต่ละ Segment มีความยาว 100 Iteration ในแต่ละ Iteration มีจำนวนคำสั่งที่จะถูกทำลาย (q) มากที่สุด 3 คำสั่ง อัตราส่วนการปรับค่าน้ำหนัก (Reaction Factor) ถูกกำหนดให้เท่ากับ 0.95 และค่าคะแนน σ_1 , σ_2 , σ_3 , และ σ_4 ถูกกำหนดให้เท่ากับ 10, 5, 3, และ 1 ตามลำดับ ส่วน Simulated Annealing ถูกกำหนดให้ยอมรับคำตอบที่มีค่า Objective ต่ำกว่าคำตอบปัจจุบันได้ไม่เกิน 50 เปอร์เซ็นต์ และมีอัตราการเย็นตัว (Cooling rate) เท่ากับ 0.99

Fail Count Limit และ Run time limit ถูกกำหนดให้เท่ากับ 800 ครั้ง และ 3600 วินาที ตามลำดับ ทั้งนี้การตั้งค่าทั้งหมดของ ALNS ในตัวอย่างการหาคำตอบด้วย ฮิวริสติกถูกสรุปไว้ในตารางที่ 3-6

ตารางที่ 3-6 การตั้งค่าของ ALNS ในตัวอย่างการหาคำตอบด้วยฮิวริสติก

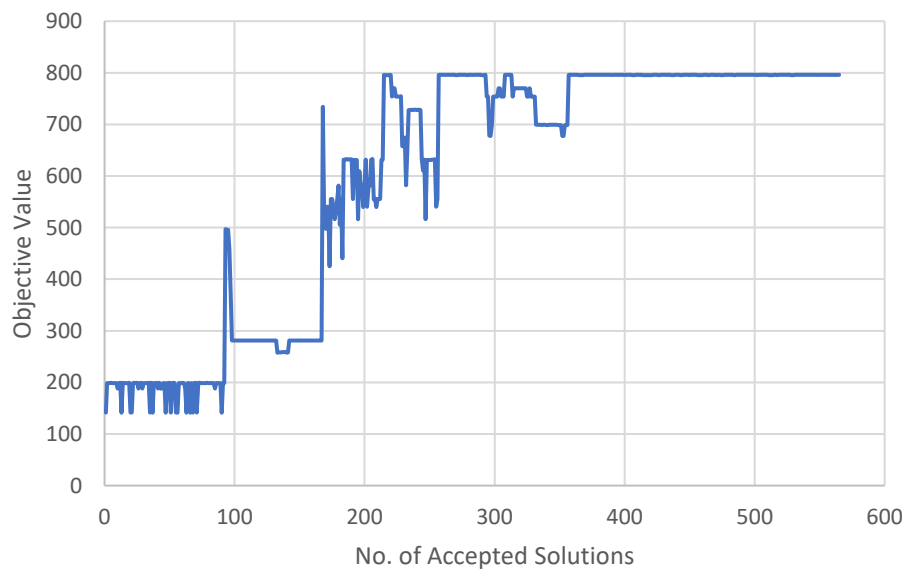
Neighborhood Size (q)	3
Max Iteration	5000
Segment Length	100
Number of Segment	50
Reaction Factor	0.95
Degrading Allowance (%)	50
Cooling rate	0.99
σ_1	10
σ_2	5
σ_3	3
σ_4	1
Fail Count Limit	800
Run Time Limit (sec)	3600

ค่า Objective ของการหาคำตอบครั้งนี้มีค่าเท่ากับ 795.86 หน่วย ซึ่งประกอบด้วยค่าตอบแทนทั้งสิ้น 803.83 หน่วย และค่าเดินทางทั้งสิ้น 7.96 หน่วย โดยใช้เวลาคำตอบ 817.63 วินาที โดยมีการปล่อยรถทั้งหมด 2 รอบ ดังแสดงในรูปที่ 3-30 เมื่อจุดสีเขียว, จุดสีดำ, และจุดสีเทา คือ ตำแหน่งท่ารถ, ตำแหน่งรับพัสดุ, และตำแหน่งส่งพัสดุ ตามลำดับ

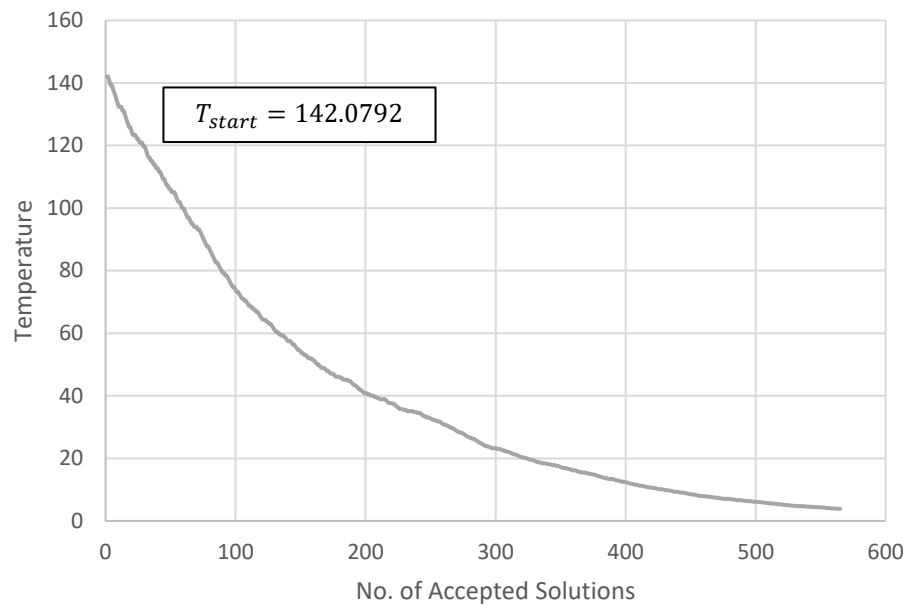


รูปที่ 3-25 ตัวอย่างคำตอบที่ได้จากฮิวริสติก

เมื่อพิจารณาคำตอบที่ถูกยอมรับโดย SA จะสังเกตเห็นว่า ค่า Objective มีแนวโน้มสูงขึ้นเรื่อย ๆ จนกระทั่ง ดังแสดงในรูปที่ 3-31 โดยในช่วงแรกของกราฟ ค่า Objective ของคำตอบที่ถูกยอมรับจะมีความผันผวนมากกว่าในช่วงท้ายของกราฟ เป็นเพราะค่าอุณหภูมิของ SA ที่เป็นตัวกำหนดค่าความน่าจะเป็นในการยอมรับคำตอบที่มีค่า Objective ต่ำกว่าคำตอบก่อนหน้ามีค่าลดลง ดังแสดงในรูปที่ 3-32



รูปที่ 3-26 ค่า Objective ของคำตอบที่ถูกยอมรับโดย SA



รูปที่ 3-27 อุณหภูมิของ SA ของรอบการหาคำตอบที่มีการยอมรับด้วย SA



บทที่ 4

ผลการดำเนินงานวิจัย

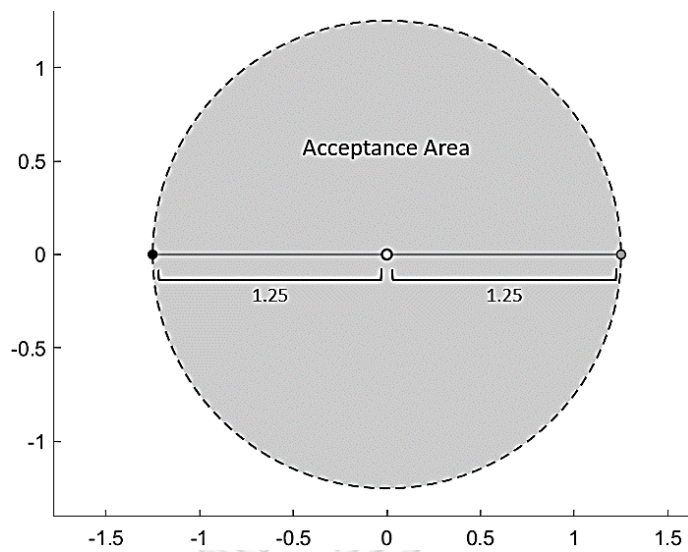
4.1 ข้อมูลนำเข้าที่ใช้ในการหาคำตอบ

ปัญหา SDPD มีข้อจำกัดด้านเวลาเป็นอย่างมาก เนื่องจากต้องให้บริการลูกค้าภายในวันทำการเดียวกัน ด้วยเหตุนี้การให้บริการรูปแบบ SDPD จึงมักให้บริการในพื้นที่ขนาดจำกัด โดยรับคำสั่งของลูกค้าในช่วงเวลาที่สามารถตอบสนองได้ทันเท่านั้น ยกตัวอย่างเช่น บริการ Bangkok Same-day ของบริษัท Kerry Express ที่ครอบคลุมแค่ภายในพื้นที่จังหวัดกรุงเทพมหานคร โดยรับคำสั่งของลูกค้าแค่ช่วงเช้าของวันทำการเท่านั้น ส่วนคำสั่งของลูกค้าที่ปรากฏหลังช่วงเวลาที่กำหนดจะถูกรวมไว้กับคำสั่งของวันทำการถัดไปแทน

เพื่อความง่ายในการจำลองปัญหา SDPD ผู้วิจัยจึงกำหนดระยะเวลาทำการเท่ากับ 10 ชั่วโมง ให้ใกล้เคียงกับระยะเวลาทำการมาตรฐานของบริษัทขนส่งโดยทั่วไป และสุ่มเวลาการปรากฏของแต่ละคำสั่งให้มีค่าอยู่ระหว่างรอบรถแรกกับรอบรถที่อยู่กึ่งกลางของระยะเวลาทำการด้วยการกระจายตัวแบบสม่ำเสมอ ($t_w \in [\frac{\omega}{2}, \omega]$)

ตำแหน่งของคำสั่งของลูกค้าถูกสุ่มให้อยู่ห่างจากท่ารถไม่เกิน 1.25 ชั่วโมง ด้วยการกระจายตัวแบบสม่ำเสมอ เพื่อให้ทุกคำสั่งมีความเป็นไปได้ที่จะได้รับบริการภายในวันทำการ ถึงแม้คำสั่งนั้นจะเป็นกรณีที่แย่ที่สุด ซึ่งเป็นการปรากฏที่กึ่งกลางของระยะเวลาทำการ และต้องใช้เวลาเดินทางทั้งสิ้น 5 ชั่วโมงเพื่อให้บริการ ดังแสดงในรูปที่ 4-1 เมื่อพื้นที่สีเทา, จุดสีขาว, จุดสีดำ, และจุดสีเทา คือขอบเขตการให้บริการ (Acceptance area), ตำแหน่งท่ารถ, ตำแหน่งรับพัสดุของคำสั่งกรณีแย่ที่สุด, และตำแหน่งส่งพัสดุของคำสั่งกรณีแย่ที่สุด ตามลำดับ

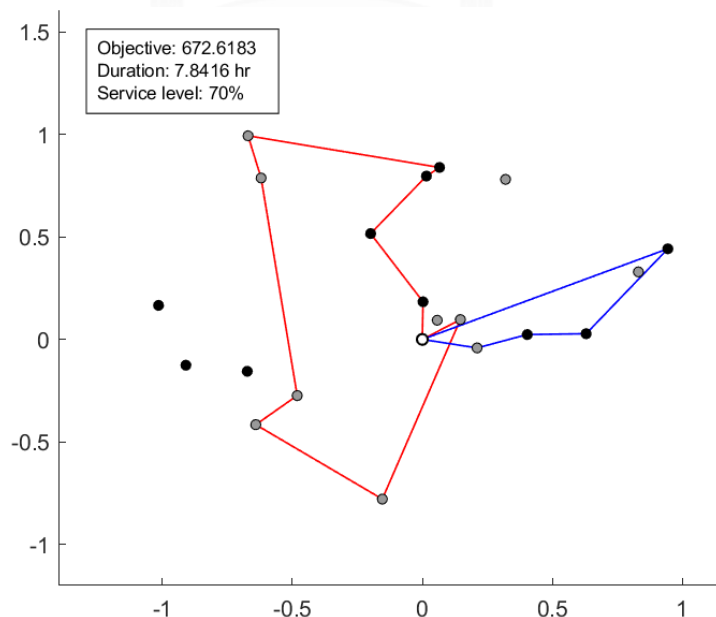
นอกจากนี้ เพื่อให้คำตอบที่ได้จากปัญหา SDPD มีระดับการให้บริการ (Service level) สูงที่สุดเท่าที่เป็นไปได้ ผู้วิจัยจึงกำหนดให้ α เท่ากับ 100 ซึ่งมีความเหมาะสมสำหรับการตั้งค่าของปัญหาในงานวิจัยนี้



รูปที่ 4-1 ขอบเขตการให้บริการ และคำสั่งกรณีที่แย่ที่สุด

4.2 การทดลองเพื่อหาขนาดปัญหาที่เหมาะสมกับสมมติฐาน

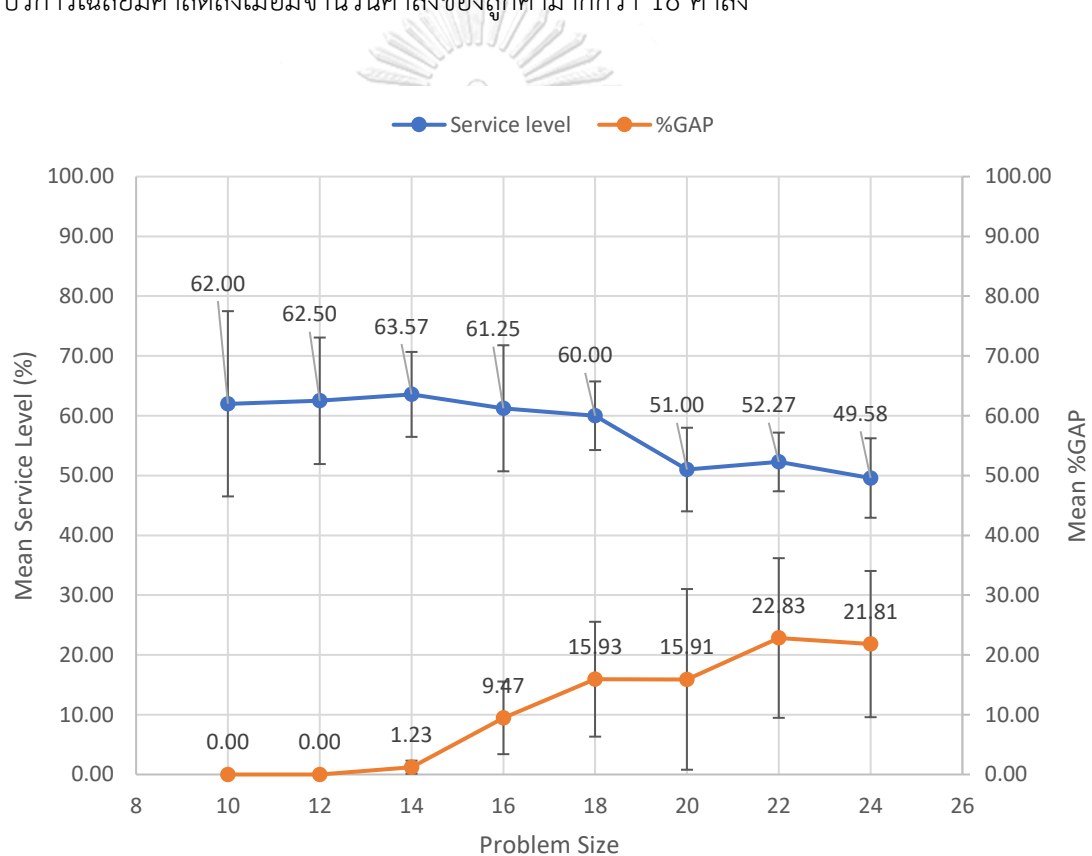
เนื่องจากสมมติฐานของปัญหากำหนดให้มีรถใช้งานเพียงคันเดียว ระดับการให้บริการจึงถูกจำกัดด้วยระยะเวลาทำการ 10 ชั่วโมง เพราะระยะเวลามากที่สุดที่รถสามารถเดินทางได้ ถูกกำหนดด้วยระยะเวลาทำการ ดังแสดงในตัวอย่างคำตอบในรูปที่ 4-2 ซึ่งรถใช้เวลาเดินทางทั้งสิ้น 7.8416 ชั่วโมง โดยมีระดับการให้บริการเท่ากับ 70 เปอร์เซ็นต์ และไม่สามารถเพิ่มคำสั่งอื่นในคำตอบได้อีก เพราะจะทำให้เกินข้อจำกัดด้านเวลา



รูปที่ 4-2 ตัวอย่างคำตอบของปัญหา SDPD ที่ระดับการให้บริการถูกจำกัดด้วยระยะเวลาทำการ

ผู้วิจัยได้ทดลองหาขนาดปัญหาที่เหมาะสม ด้วยการสุ่มตัวอย่างข้อมูลนำเข้าที่มี 6 รอบรถ และมีจำนวนคำสั่งของลูกค้า 10, 12, 14, 16, 18, 20, 22, และ 24 คำสั่ง ตามลำดับ อย่างละ 10 ตัวอย่าง เพื่อหาคำตอบที่ดีที่สุดด้วยวิธีแมนตรง ในระยะเวลาหาคำตอบ 4 ชั่วโมง ด้วยคอมพิวเตอร์ที่มี CPU Intel Core i5-4570 และแรม 8 GB โดยใช้โปรแกรม IBM ILOG CPLEX Optimization Studio จากนั้นจึงบันทึกระดับการให้บริการ, เวลาที่ใช้ในการหาคำตอบ (CPU time), และ %GAP¹ หากไม่สามารถหาคำตอบที่ดีที่สุดได้ภายในระยะเวลาที่กำหนด

ผลการทดลองดังแสดงในรูปที่ 4-3 แสดงให้เห็นว่า แม้ปัญหา SDPD จะมีจำนวนคำสั่งของลูกค้าเพียง 10 คำสั่ง แต่ระดับการให้บริการเฉลี่ยมีค่าเพียง 62 เปอร์เซ็นต์เท่านั้น โดยที่ระดับการให้บริการเฉลี่ยมีค่าลดลงเมื่อมีจำนวนคำสั่งของลูกค้ามากกว่า 18 คำสั่ง



รูปที่ 4-3 ระดับการให้บริการเฉลี่ย และ %GAP เฉลี่ยของคำตอบจากปัญหา SDPD ที่มี 6 รอบรถ

¹ %GAP คือ เปอร์เซ็นต์ความแตกต่างระหว่างคำตอบที่ดีที่สุดที่หาได้ และค่าที่เชื่อมโยงกับคำตอบที่ดีที่สุดที่เป็นไปได้

ความสามารถในการหาคำตอบด้วยวิธีแมนตรง มีแนวโน้มลดลงตามขนาดปัญหาที่เพิ่มขึ้น โดยสังเกตได้จากการเพิ่มขึ้นของ %GAP เฉลี่ย เมื่อเพิ่มขนาดของปัญหา ดังแสดงในรูปที่ 4-3 ทั้งนี้ โปรแกรม CPLEX ไม่สามารถหาคำตอบที่ดีที่สุดได้ภายในระยะเวลาหาคำตอบที่กำหนดในบางกลุ่ม ตัวอย่าง ดังที่ได้สรุปไว้ในตารางที่ 4-1 ซึ่งมีสาเหตุจากการที่จำนวนตัวแปรตัดสินใจ y_i^w , x_a^w , และ k_i ในแบบจำลองทางคณิตศาสตร์มีจำนวนเพิ่มสูงอย่างรวดเร็วเมื่อเพิ่มจำนวนคำสั่งของลูกค้าเพียงเล็กน้อย

ตารางที่ 4-1 เวลาในการหาคำตอบในหน่วยวินาทีของปัญหามาขนาดต่างกันที่มี 6 รอบรถ (วินาที)

Size	Instance									
	1	2	3	4	5	6	7	8	9	10
10	43	1	4	1	9	3	92	7	123	17
12	13	187	29	58	286	27	477	108	152	139
14	214	14400*	452	2851	392	14400*	520	1748	313	113
16	273	14401*	14401*	14400*	3402	1174	1463	14401*	2342	3730
18	14373	14405*	14402*	4482	14407*	3851	1757	14400*	14401*	2979
20	14441*	14400*	14436*	14401*	6271	14401*	14400*	14401*	14401*	14401*
22	4113	12304	14404*	14400*	7037	14401*	14403*	14402*	14400*	14409*
24	14404*	14401*	14407*	14404*	14440*	14401*	14410*	14401*	14400*	14401*

* ไม่สามารถหาคำตอบที่ดีที่สุดได้ภายในระยะเวลา 4 ชั่วโมง

ตัวอย่างที่ไม่สามารถหาคำตอบที่ดีที่สุดได้ภายในระยะเวลา 4 ชั่วโมง จะได้คำตอบที่มีคุณภาพขึ้นกับ %GAP แทน ดังสรุปไว้ในตารางที่ 4-2 เมื่อช่องสีทึบในตาราง คือ ตัวอย่างที่สามารถหาคำตอบที่ดีที่สุดได้ (%GAP = 0)

ตารางที่ 4-2 %GAP ของการหาคำตอบของปัญหามาขนาดต่างกันที่มี 6 รอบรถ

Size	Instance									
	1	2	3	4	5	6	7	8	9	10
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.46	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00
16	0.00	12.04	16.57	2.77	0.00	0.00	0.00	6.50	0.00	0.00
18	0.00	13.83	5.91	0.00	31.78	0.00	0.00	15.76	12.38	0.00
20	20.03	3.03	30.42	13.22	0.00	9.13	0.97	48.44	9.85	8.08
22	0.00	0.00	20.88	5.41	0.00	29.54	38.83	34.87	5.18	25.08
24	5.67	13.87	37.89	27.48	33.51	18.83	39.47	20.60	7.97	12.84

สำหรับขนาดของปัญหาที่เหมาะสมในการทดสอบประสิทธิภาพของฮิวริสติกส์ที่พัฒนาขึ้นนั้น ผู้วิจัยได้เลือกเอากลุ่มปัญหาที่มีจำนวนคำสั่งของลูกค้า 16 คำสั่ง ซึ่งมี %GAP เฉลี่ยเท่ากับ 9.47 เปอร์เซ็นต์ เนื่องจากปัญหาในกลุ่มดังกล่าวมี %GAP เฉลี่ยที่ไม่สูง หรือต่ำเกินไป อีกทั้งยังมีโอกาสที่ CPLEX สามารถหาคำตอบที่เหมาะสมได้ในขอบเขตของเวลาที่กำหนด

4.3 การทดลองเพื่อหาขีดจำกัดความสามารถของวิธีแมนตรง

การเพิ่มจำนวนรอบรถในวันทำการ (ω) สามารถเพิ่มประสิทธิภาพของการดำเนินงานในปัญหา SDPD ได้ เนื่องจากการมีจำนวนรอบรถมากขึ้นทำให้มีจุดตัดสินใจ (Decision epochs) ในการวางแผนเพิ่มขึ้น แต่เนื่องจากตัวแปรตัดสินใจได้แก่ y_i^w , x_a^w , v_k^w , และ z_k มีจำนวนเพิ่มขึ้นตามค่าของตัวแปร ω ดังนั้นความสามารถในการหาคำตอบที่ดีที่สุดของปัญหา SDPD จึงลดลงเมื่อเพิ่มจำนวนรอบรถในวันทำการ

เพื่อทดลองหาขีดจำกัดความสามารถของการหาคำตอบด้วยวิธีแมนตรงของปัญหา SDPD ผู้วิจัยได้นำตัวอย่างที่มีจำนวนคำสั่งของลูกค้า 16 คำสั่ง ทั้ง 10 ตัวอย่างจากการทดลอง 4.2 มาแปรจำนวนรอบรถในวันทำการทั้งหมด 4 ระดับ คือ 6, 8, 10, และ 12 รอบ ตามลำดับ ผ่านวิธีการหาคำตอบด้วยวิธีแมนตรง ภายในระยะเวลาหาคำตอบ 4 ชั่วโมง ด้วยคอมพิวเตอร์ที่มี CPU Intel Core i5-4570 และแรม 8 GB โดยใช้โปรแกรม IBM ILOG CPLEX Optimization Studio จากนั้นจึงบันทึกเวลาที่ใช้ในการหาคำตอบ (CPU time), และ %GAP หากไม่สามารถหาคำตอบที่ดีที่สุดได้ภายในระยะเวลาที่กำหนด

ผลการทดลองดังสรุปไว้ในตารางที่ 4-3 แสดงให้เห็นว่า ความสามารถในการหาคำตอบด้วยวิธีแมนตรง มีแนวโน้มลดลงตามจำนวนรอบรถต่อวันที่เพิ่มขึ้น โดยมีตัวอย่างจำนวนมาก ที่โปรแกรม CPLEX ไม่สามารถหาคำตอบที่ดีที่สุดได้ภายในระยะเวลาหาคำตอบที่กำหนด

ตารางที่ 4-3 เวลาในการหาคำตอบของปัญหาขนาด 16 คำสั่งด้วยวิธีแมนตรง (วินาที)

ω	Instance										Average
	1	2	3	4	5	6	7	8	9	10	
6	273	14401*	14401*	14400*	3402	1174	1463	14401*	2342	3730	6999
8	2011	14410*	14400*	14402*	14401*	4704	6188	14403*	14400*	8385	10770
10	9763	14403*	14401*	14401*	14406*	14400*	12474	14426*	14401*	14401*	13748
12	14401*	14401*	14401*	14401*	14407*	14409*	14404*	14401*	14401*	14404*	14403

* ไม่สามารถหาคำตอบที่ดีที่สุดได้ภายในระยะเวลา 4 ชั่วโมง

ตัวอย่างที่ไม่สามารถหาคำตอบที่ดีที่สุดได้ภายในระยะเวลา 4 ชั่วโมง จะได้คำตอบที่มีคุณภาพขึ้นกับ %GAP ดังสรุปไว้ในตารางที่ 4-4 เมื่อช่องสี่ทึบในตาราง คือ ตัวอย่างที่สามารถหาคำตอบที่ดีที่สุดได้ (%GAP = 0)

ตารางที่ 4-4 %GAP ของการหาคำตอบของปัญหาขนาด 16 คำสั่งที่ด้วยวิธีแมนตรง

ω	Instance										Average
	1	2	3	4	5	6	7	8	9	10	
6	0.00	12.04	16.57	2.77	0.00	0.00	0.00	6.50	0.00	0.00	3.79
8	0.00	24.14	17.76	14.50	8.79	0.00	0.00	22.77	19.31	0.00	10.73
10	0.00	37.55	13.39	23.62	25.07	10.84	0.00	31.03	29.85	16.24	18.76
12	9.55	49.96	25.33	35.94	27.85	40.90	19.97	23.47	29.42	17.54	27.99

การทดลองเพื่อหาขีดจำกัดความสามารถของการหาคำตอบด้วยวิธีแมนตรง ทำให้ได้ข้อสรุปว่า การเพิ่มจำนวนรอบรถต่อวัน ยิ่งทำให้ประสิทธิภาพการหาคำตอบด้วยวิธีแมนตรง น้อยลง สังเกตจากค่าเฉลี่ย %GAP ที่เพิ่มขึ้นระหว่างกลุ่มตัวอย่าง โดยกลุ่มตัวอย่างที่มี 6, 8, 10, และ 12 รอบรถ มีค่าเฉลี่ย %GAP เท่ากับ 3.79, 10.73, 18.76, และ 27.99 เปอร์เซ็นต์ ตามลำดับ

4.4 การตั้งค่าของฮิวริสติก

จากการทดลองเพื่อหาคำตอบของปัญหา SDPD ที่มีจำนวนคำสั่งของลูกค้าทั้งหมด 16 คำสั่ง ด้วยวิธีฮิวริสติกเบื้องต้น ผู้วิจัยได้กำหนดให้ ALNS มีจำนวนรอบการค้นหาคำตอบเท่ากับ 10000 รอบ โดยแต่ละ Segment มีความยาวเท่ากับ 20 รอบ และในแต่ละรอบมีจำนวนคำสั่งที่จะถูกทำลาย (q) เท่ากับ 3 คำสั่ง ซึ่งการตั้งค่าดังกล่าวจะส่งผลทำให้ฮิวริสติกใช้เวลาเฉลี่ยในการหาคำตอบไม่เกิน 600 วินาที

เมื่อสิ้นสุด Segment ความน่าจะเป็นในการเลือกวิธีทำลาย และวิธีซ่อมแซมคำตอบจะถูกปรับปรุงด้วยอัตราส่วนการปรับค่าน้ำหนัก (Reaction Factor) เท่ากับ 0.80 เพื่อให้ผลลัพธ์ลู่เข้าสู่ Local optimum ไม่ช้า หรือเร็วเกินไป ส่วนค่า σ_1 , σ_2 , σ_3 , และ σ_4 ถูกกำหนดให้มีค่าเท่ากับ 15, 5, 3, และ 1 ตามลำดับ ซึ่งให้ค่าน้ำหนักกับวิธีการทำลาย และซ่อมแซมคำตอบที่ดีที่สุด ในขณะที่ยังคงให้ค่าน้ำหนักกับวิธีการทำลาย และซ่อมแซมคำตอบที่ไม่ก่อให้เกิดการปรับปรุงค่า Objective value เพื่อหลีกเลี่ยงการใช้เรียกใช้วิธีการทำลาย และซ่อมแซมคำตอบซ้ำๆ ในรอบการปรับปรุงการตอบอื่น

คำตอบใหม่ที่เกิดขึ้นในแต่ละรอบที่มีค่า Objective ต่ำกว่าคำตอบปัจจุบันไม่เกิน 50 เปอร์เซ็นต์ อาจถูก Simulated Annealing ยอมรับด้วยความน่าจะเป็นที่ขึ้นกับอุณหภูมิ ซึ่งอุณหภูมิดังกล่าวจะลดลงด้วยอัตราการเย็นตัวเท่ากับ 0.80 จนมีค่าเข้าใกล้ศูนย์ภายในระยะเวลาหาคำตอบที่คาดหวังไว้ ทั้งนี้การตั้งค่าของฮิวริสติกถูกสรุปไว้ในตารางที่ 4-5

ตารางที่ 4-5 การตั้งค่าของฮิวริสติก

Neighborhood Size (q)	3
Max Iteration	10000
Segment Length	20
Number of Segment	500
Reaction Factor	0.80
Degrading Allowance (%)	50
Cooling rate	0.80
σ_1	10
σ_2	5
σ_3	3
σ_4	1
Fail Count Limit	200
Run Time Limit (sec)	1200

4.5 ผลการเปรียบเทียบคุณภาพของคำตอบ และเวลาในการหาคำตอบของฮิวริสติก

ประสิทธิภาพในการหาคำตอบของฮิวริสติกในงานวิจัยนี้ ทั้งในด้านคุณภาพของคำตอบ และเวลาในการหาคำตอบถูกประเมินโดยนำคำตอบที่ได้จากฮิวริสติกมาเปรียบเทียบกับคำตอบที่ได้จากวิธีแมนตรง จากการทดลองในหัวข้อที่ 4.3 โดยผู้วิจัยได้หาคำตอบด้วยฮิวริสติกเป็นจำนวน 10 ครั้งต่อ 1 ตัวอย่าง ด้วยคอมพิวเตอร์ที่มี CPU Intel Core i5-4570 และแรม 8 GB โดยใช้โปรแกรม MATLAB R2019a จากนั้นจึงทำการบันทึกคำตอบ และเวลาที่ใช้ในการหาคำตอบแต่ละครั้ง

4.5.1 ประสิทธิภาพของฮิวริสติกในการหาคำตอบของปัญหาที่มี 6 รอบรถ

จากการเปรียบเทียบคำตอบที่ได้จากฮิวริสติก และวิธีแมนตรง พบว่ามี 1 ตัวอย่าง คือ ตัวอย่างที่ 3 ที่คุณภาพของคำตอบที่ได้จากฮิวริสติกดีกว่าคำตอบที่ได้จากวิธีแมนตรง โดยมีส่วนต่างเท่ากับ 6.91 เปอร์เซ็นต์ นอกจากนี้ยังพบว่ามีทั้งหมด 8 ตัวอย่าง ได้แก่ ตัวอย่างที่ 1, 2, 4, 5, 6, 7, 8, และ 9 ที่คุณภาพของคำตอบที่ได้จากฮิวริสติกเทียบเท่าคำตอบที่ได้จากวิธีแมนตรง และพบว่ามี 1 ตัวอย่าง คือ ตัวอย่างที่ 10 ที่คุณภาพของคำตอบที่ได้จากฮิวริสติกต่ำกว่าคำตอบที่ได้จากวิธีแมนตรง โดยมีส่วนต่างเท่ากับ 1.27 เปอร์เซ็นต์ ทั้งนี้คำตอบทั้งหมดที่ได้จากการหาคำตอบด้วยฮิวริสติกถูกสรุปไว้ในตารางที่ 4-6

ตารางที่ 4-6 ผลการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 6 รอบรถ

Solver	Parameter	Instance									
		1	2	3	4	5	6	7	8	9	10
CPLEX	Objective	797.50	986.55	831.77	749.80	833.27	806.03	820.33	922.21	946.35	932.75
	%GAP	0.00	12.04	16.57	2.77	0.00	0.00	0.00	6.50	0.00	0.00
ALNS	Replicate 1	797.50	986.55	889.22	749.80	770.22	806.03	820.31	820.68	791.23	828.34
	Replicate 2	797.50	986.55	889.22	749.80	833.27	794.21	820.31	820.68	791.23	827.16
	Replicate 3	797.50	973.85	887.80	749.80	770.22	779.48	809.66	850.28	946.35	920.90
	Replicate 4	797.50	986.55	887.43	749.80	833.27	779.48	809.66	879.47	946.35	814.23
	Replicate 5	797.50	986.55	887.43	749.80	833.27	806.03	820.33	922.21	870.19	901.37
	Replicate 6	797.50	986.55	874.35	739.28	833.27	806.03	809.73	879.21	946.35	920.90
	Replicate 7	797.50	986.55	887.70	749.80	833.27	791.48	809.73	873.53	946.35	920.90
	Replicate 8	797.50	986.55	887.80	702.95	833.27	779.48	820.31	823.83	946.35	814.23
	Replicate 9	797.50	986.55	887.80	749.80	770.22	806.03	809.73	846.81	946.35	901.37
	Replicate 10	797.50	986.55	889.22	749.80	770.22	779.48	809.73	846.46	946.35	901.37
	Average	Objective	797.50	985.28	886.80	744.06	808.05	792.77	813.95	856.31	907.71
S.D.		0.00	4.02	4.44	14.82	32.56	12.53	5.48	32.52	65.82	47.44
	Different (%)	0.00	-0.13	6.62	-0.77	-3.03	-1.64	-0.78	-7.15	-4.08	-6.18
Best	Objective	797.50	986.55	889.22	749.80	833.27	806.03	820.33	922.21	946.35	920.90
	Different (%)	0.00	0.00	6.91	0.00	0.00	0.00	0.00	0.00	0.00	-1.27

จากการเปรียบเทียบเวลาเฉลี่ยในการหาคำตอบของฮิวริสติก และวิธีแมนตรง พบว่าฮิวริสติกใช้เวลาเฉลี่ยในการหาคำตอบน้อยกว่าวิธีแมนตรง ในทุกตัวอย่าง ยกเว้นตัวอย่างที่ 1 ซึ่งฮิวริสติกใช้เวลาเฉลี่ยในการหาคำตอบมากกว่าวิธีแมนตรง 9.14 เปอร์เซ็นต์ โดยฮิวริสติกใช้เวลาเฉลี่ยในการหาคำตอบทั้งหมดเท่ากับ 420.29 วินาที (ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 135.24) ในขณะที่วิธีแมนตรงใช้เวลาเฉลี่ยในการหาคำตอบทั้งหมดเท่ากับ 6998.67 วินาที (ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 6448.75) ทั้งนี้เวลาในการหาคำตอบทั้งหมดที่ได้จากการหาคำตอบด้วยฮิวริสติกในหน่วยวินาทีถูกสรุปไว้ในตารางที่ 4-7

ตารางที่ 4-7 เวลาในการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 6 รอบรถ (วินาที)

Solver		Instance										
		1	2	3	4	5	6	7	8	9	10	
CPLEX		272.65	14401.31	14400.72	14400.23	3402.18	1173.96	1462.63	14400.68	2341.89	3730.48	
ALNS	Replicate 1	294.81	292.78	530.94	419.65	241.50	638.43	431.23	300.07	368.64	295.89	
	Replicate 2	304.29	290.17	593.88	767.26	450.29	294.72	671.92	469.86	351.64	364.59	
	Replicate 3	345.34	362.10	411.03	440.67	243.07	257.39	342.02	700.37	373.29	544.00	
	Replicate 4	316.81	333.69	728.66	445.74	433.00	242.12	483.19	276.78	489.79	364.17	
	Replicate 5	249.51	319.69	372.26	402.92	334.79	569.90	531.74	415.20	300.92	457.67	
	Replicate 6	314.79	322.97	659.04	535.37	625.25	417.66	412.41	283.33	628.45	642.48	
	Replicate 7	344.56	345.21	360.43	503.85	541.60	528.35	319.36	470.33	361.74	381.66	
	Replicate 8	291.76	517.32	327.29	425.33	634.62	257.88	408.16	572.71	391.35	371.89	
	Replicate 9	223.13	319.00	289.26	698.11	239.84	512.48	453.80	740.08	359.89	484.08	
	Replicate 10	290.59	307.89	601.79	402.44	274.52	307.80	348.91	724.46	378.29	414.98	
	Average		297.56	341.08	487.46	504.14	401.85	402.67	440.27	495.32	400.40	432.14
	S.D.		38.21	65.72	154.23	128.76	158.05	149.12	104.65	182.26	92.93	102.37
Different (%)		9.14	-97.63	-96.62	-96.50	-88.19	-65.70	-69.90	-96.56	-82.90	-88.42	

4.5.2 ประสิทธิภาพของฮิวริสติกในการหาคำตอบของปัญหาที่มี 8 รอบรถ

จากการเปรียบเทียบคำตอบที่ได้จากฮิวริสติก และวิธีแมนตรง พบว่ามี 5 ตัวอย่าง ได้แก่ ตัวอย่างที่ 2, 3, 5, 8, และ 9 ที่คุณภาพของคำตอบที่ได้จากฮิวริสติกดีกว่าคำตอบที่ได้จากวิธีแมนตรง โดยมีส่วนต่างเท่ากับ 0.98, 8.34, 0.03, 4.14, และ 4.78 เปอร์เซ็นต์ ตามลำดับ นอกจากนี้ยังพบว่ามีทั้งหมด 4 ตัวอย่าง ได้แก่ ตัวอย่างที่ 1, 4, 6, และ 7 ที่คุณภาพของคำตอบที่ได้จากฮิวริสติกเทียบเท่าคำตอบที่ได้จากวิธีแมนตรง และพบว่ามี 1 ตัวอย่าง คือ ตัวอย่างที่ 10 ที่คุณภาพของคำตอบที่ได้จากฮิวริสติกต่ำกว่าคำตอบที่ได้จากวิธีแมนตรง โดยมีส่วนต่างเท่ากับ 1.20 เปอร์เซ็นต์ ทั้งนี้คำตอบทั้งหมดที่ได้จากการหาคำตอบด้วยฮิวริสติกถูกสรุปไว้ในตารางที่ 4-8

จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 4-8 ผลการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 8 รอบรถ

Solver	Parameter	Instance										
		1	2	3	4	5	6	7	8	9	10	
CPLEX	Objective	810.28	979.81	831.73	758.10	791.93	859.17	820.33	843.06	851.45	945.20	
	%GAP	0.00	24.14	17.76	14.50	8.79	0.00	0.00	22.77	19.31	0.00	
ALNS	Replicate 1	810.28	985.95	900.89	758.10	744.01	800.84	757.90	798.71	892.16	856.97	
	Replicate 2	735.05	985.95	900.89	758.10	791.93	839.28	820.31	798.46	892.16	933.83	
	Replicate 3	810.28	989.44	900.99	739.28	725.79	800.84	820.33	823.30	827.68	933.83	
	Replicate 4	810.28	989.44	901.06	758.10	745.57	839.28	820.31	877.95	878.01	802.14	
	Replicate 5	735.05	989.44	887.31	749.80	792.17	839.28	774.77	798.46	878.01	933.83	
	Replicate 6	735.05	985.95	887.31	758.10	744.01	812.51	810.15	798.46	892.16	933.83	
	Replicate 7	735.05	989.44	887.31	758.10	791.93	839.28	820.33	795.47	763.38	933.83	
	Replicate 8	810.28	989.44	888.40	758.10	791.93	843.59	774.77	843.06	827.68	933.83	
	Replicate 9	810.28	985.95	887.70	758.10	778.59	800.84	774.77	843.06	878.01	802.14	
	Replicate 10	810.28	985.95	901.06	758.10	791.93	859.19	820.01	797.31	878.01	933.83	
	Average	Objective	780.19	987.69	894.29	755.39	769.79	827.49	799.37	817.43	860.73	899.81
	S.D.		38.85	1.84	7.06	6.23	26.65	21.54	25.46	28.53	41.92	56.78
Different (%)		-3.71	0.80	7.52	-0.36	-2.80	-3.69	-2.56	-3.04	1.09	-4.80	
Best	Objective	810.28	989.44	901.06	758.10	792.17	859.19	820.33	877.95	892.16	933.83	
	Different (%)	0.00	0.98	8.34	0.00	0.03	0.00	0.00	4.14	4.78	-1.20	

จากการเปรียบเทียบเวลาเฉลี่ยในการหาคำตอบของฮิวริสติก และวิธีแมนตรง พบว่าฮิวริสติกใช้เวลาเฉลี่ยในการหาคำตอบน้อยกว่าวิธีแมนตรง ในทุกตัวอย่าง โดยฮิวริสติกใช้เวลาเฉลี่ยในการหาคำตอบทั้งหมดเท่ากับ 474.49 วินาที (ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 126.88) ในขณะที่วิธีแมนตรงใช้เวลาเฉลี่ยในการหาคำตอบทั้งหมดเท่ากับ 10770.44 วินาที (ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 4937.26) ซึ่งเวลาในการหาคำตอบทั้งหมดที่ได้จากการหาคำตอบด้วยฮิวริสติกในหน่วยวินาทีถูกสรุปไว้ในตารางที่ 4-9

ตารางที่ 4-9 เวลาในการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 8 รอบรถ (วินาที)

Solver		Instance									
		1	2	3	4	5	6	7	8	9	10
CPLEX		2010.87	14409.53	14400.46	14402.28	14400.71	4704.13	6187.55	14403.49	14400.43	8384.92
ALNS	Replicate 1	487.35	459.36	350.73	426.34	421.04	449.48	491.70	614.31	427.45	346.29
	Replicate 2	354.35	327.36	490.10	492.73	422.65	482.13	1024.24	448.00	529.79	427.20
	Replicate 3	375.47	369.21	388.33	578.97	396.52	391.61	741.07	427.68	382.45	391.89
	Replicate 4	567.17	389.33	440.52	430.95	409.57	434.39	526.26	479.24	449.58	355.42
	Replicate 5	351.14	492.94	388.50	456.43	616.25	436.41	465.54	467.72	307.95	409.62
	Replicate 6	371.09	397.88	641.63	504.16	503.06	397.97	637.61	514.25	488.16	493.03
	Replicate 7	316.89	463.90	446.35	493.96	419.47	718.83	585.39	721.98	749.92	507.36
	Replicate 8	481.08	586.13	320.16	518.49	981.98	451.31	428.38	458.09	407.75	338.37
	Replicate 9	480.73	545.64	358.95	445.42	514.71	443.40	358.48	623.50	301.98	337.76
	Replicate 10	496.18	611.89	436.71	512.72	388.14	809.66	456.59	488.18	299.03	403.61
	Average	428.15	464.36	426.20	486.02	507.34	501.52	571.53	524.30	434.41	401.06
S.D.	83.54	95.40	91.61	47.10	181.23	142.48	193.51	96.15	136.25	61.12	
Different (%)	-78.71	-96.78	-97.04	-96.63	-96.48	-89.34	-90.76	-96.36	-96.98	-95.22	

4.5.3 ประสิทธิภาพของฮิวริสติกในการหาคำตอบของปัญหาที่มี 10 รอบรถ

จากการเปรียบเทียบคำตอบที่ได้จากฮิวริสติก และวิธีแมนตรง พบว่ามี 6 ตัวอย่าง ได้แก่ ตัวอย่างที่ 2, 3, 5, 8, 9, และ 10 ที่คุณภาพของคำตอบที่ได้จากฮิวริสติกดีกว่าคำตอบที่ได้จากวิธีแมนตรง โดยมีส่วนต่างเท่ากับ 14.21, 0.08, 6.04, 12.38, 27.55, และ 2.71 เปอร์เซ็นต์ ตามลำดับ นอกจากนี้ยังพบว่ามีทั้งหมด 3 ตัวอย่าง ได้แก่ ตัวอย่างที่ 1, 4, และ 7 ที่คุณภาพของคำตอบที่ได้จากฮิวริสติกเทียบเท่ากับคำตอบที่ได้จากวิธีแมนตรง และพบว่ามี 1 ตัวอย่าง คือ ตัวอย่างที่ 7 ที่คุณภาพของคำตอบที่ได้จากฮิวริสติกต่ำกว่าคำตอบที่ได้จากวิธีแมนตรง โดยมีส่วนต่างเท่ากับ 1.62 เปอร์เซ็นต์ ทั้งนี้คำตอบทั้งหมดที่ได้จากการหาคำตอบด้วยฮิวริสติกถูกสรุปไว้ในตารางที่ 4-10

ตารางที่ 4-10 ผลการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 10 รอบรถ

Solver	Parameter	Instance										
		1	2	3	4	5	6	7	8	9	10	
CPLEX	Objective	886.28	863.83	888.94	758.41	785.67	908.75	862.74	795.96	768.47	897.03	
	%GAP	0.00	37.55	13.39	23.62	25.07	10.84	0.00	31.03	29.85	16.24	
ALNS	Replicate 1	886.28	986.56	846.01	758.41	791.88	861.89	862.53	823.83	980.18	892.59	
	Replicate 2	812.67	986.56	846.01	758.41	763.18	894.01	862.53	849.99	980.18	921.35	
	Replicate 3	797.13	986.56	875.54	758.41	833.12	698.55	862.74	823.83	980.18	892.59	
	Replicate 4	797.13	986.56	865.62	758.41	832.70	808.88	862.74	798.58	980.18	892.59	
	Replicate 5	797.13	986.56	887.36	739.28	806.54	873.70	740.01	774.53	980.18	789.77	
	Replicate 6	797.13	986.56	887.36	657.95	792.31	779.17	740.16	894.49	828.42	813.99	
	Replicate 7	797.13	986.56	875.04	661.65	791.88	861.89	862.74	796.08	980.18	792.10	
	Replicate 8	886.28	986.56	889.66	758.41	807.90	861.89	862.53	829.29	824.69	892.59	
	Replicate 9	797.26	986.56	875.54	758.41	778.84	808.88	862.74	798.58	824.69	892.59	
	Replicate 10	886.28	985.53	887.48	758.41	744.26	873.70	862.53	823.83	980.18	791.87	
	Average	Objective	825.44	986.46	873.56	736.78	794.26	832.26	838.13	821.30	933.90	857.20
		S.D.	42.25	0.33	16.41	41.02	27.97	59.32	51.67	33.53	74.51	53.02
		Different (%)	-6.86	14.20	-1.73	-2.85	1.09	-8.42	-2.85	3.18	21.53	-4.44
Best	Objective	886.28	986.56	889.66	758.41	833.12	894.01	862.74	894.49	980.18	921.35	
	Different (%)	0.00	14.21	0.08	0.00	6.04	-1.62	0.00	12.38	27.55	2.71	

จากการเปรียบเทียบเวลาเฉลี่ยในการหาคำตอบของฮิวริสติก และวิธีแมนตรง พบว่าฮิวริสติกใช้เวลาเฉลี่ยในการหาคำตอบน้อยกว่าวิธีแมนตรง ในทุกตัวอย่าง โดยฮิวริสติกใช้เวลาเฉลี่ยในการหาคำตอบทั้งหมดเท่ากับ 537.61 วินาที (ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 159.33) ในขณะที่วิธีแมนตรงใช้เวลาเฉลี่ยในการหาคำตอบทั้งหมดเท่ากับ 13747.54 วินาที (ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1525.95) ซึ่งเวลาในการหาคำตอบทั้งหมดที่ได้จากการหาคำตอบด้วยฮิวริสติกในหน่วยวินาทีถูกสรุปไว้ในตารางที่ 4-11

ตารางที่ 4-11 เวลาในการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 10 รอบรถ (วินาที)

Solver	Parameter	Instance										
		1	2	3	4	5	6	7	8	9	10	
CPLEX		9762.79	14402.59	14400.61	14400.96	14405.96	14400.38	12474.08	14426.29	14400.55	14401.20	
ALNS	Replicate 1	418.47	645.54	535.92	422.87	1200.54	466.18	655.52	435.09	545.33	576.34	
	Replicate 2	431.90	485.98	443.83	493.54	464.88	486.63	503.87	797.15	631.51	1125.81	
	Replicate 3	350.82	593.50	660.46	397.87	653.65	378.62	420.48	431.28	519.14	491.70	
	Replicate 4	401.06	708.47	534.24	458.74	1113.38	306.65	495.30	613.48	504.86	436.72	
	Replicate 5	360.14	456.24	680.48	720.62	767.84	462.35	557.20	543.78	471.70	457.17	
	Replicate 6	378.62	596.20	423.86	435.79	472.66	295.96	579.61	456.71	420.25	614.10	
	Replicate 7	377.80	561.90	530.84	415.60	579.40	566.27	624.22	557.92	787.94	421.65	
	Replicate 8	703.97	553.44	442.14	513.61	553.41	562.88	813.84	405.12	467.41	410.94	
	Replicate 9	397.36	612.46	657.79	409.73	446.98	364.99	463.35	971.28	386.73	468.77	
	Replicate 10	602.06	443.99	446.50	622.50	441.49	493.68	535.01	636.22	547.54	574.03	
		Average	442.22	565.77	535.61	489.09	669.42	438.42	564.84	584.80	528.24	557.72
		S.D.	116.30	84.30	99.51	105.51	277.42	97.29	112.85	181.05	114.50	211.76
		Different (%)	-95.47	-96.07	-96.28	-96.60	-95.35	-96.96	-95.47	-95.95	-96.33	-96.13

4.5.4 ประสิทธิภาพของฮิวริสติกในการหาคำตอบของปัญหาที่มี 12 รอบรถ

จากการเปรียบเทียบคำตอบที่ได้จากฮิวริสติก และวิธีแมนตรง พบว่ามี 8 ตัวอย่าง ได้แก่ ตัวอย่างที่ 2,3,4,5,6,7,9 และ 10 ที่คุณภาพของคำตอบที่ได้จากฮิวริสติกดีกว่าคำตอบที่ได้จากวิธีแมนตรง โดยมีส่วนต่างเท่ากับ 28.05, 10.48, 9.86, 9.18, 21.93, 2.07, 5.98, และ 6.11 เปอร์เซ็นต์ ตามลำดับ นอกจากนี้ยังพบว่ามี 1 ตัวอย่าง คือ ตัวอย่างที่ 1 ที่คุณภาพของคำตอบที่ได้จากฮิวริสติกเทียบเท่าคำตอบที่ได้จากวิธีแมนตรง และพบว่ามี 1 ตัวอย่าง คือ ตัวอย่างที่ 8 ที่คุณภาพของคำตอบที่ได้จากฮิวริสติกต่ำกว่าคำตอบที่ได้จากวิธีแมนตรง โดยมีส่วนต่างเท่ากับ 5.66 เปอร์เซ็นต์ ทั้งนี้คำตอบทั้งหมดที่ได้จากการหาคำตอบด้วยฮิวริสติกถูกรวบรวมไว้ในตารางที่ 4-12

ตารางที่ 4-12 ผลการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 12 รอบรถ

Solver	Parameter	Instance									
		1	2	3	4	5	6	7	8	9	10
CPLEX	Objective	886.28	772.68	816.92	690.03	792.17	744.70	809.78	946.05	892.91	920.12
	%GAP	9.55	49.96	25.33	35.94	27.85	40.90	19.97	23.47	29.42	17.54
ALNS	Replicate 1	886.28	986.55	887.43	757.40	772.27	829.36	820.31	874.06	892.91	857.17
	Replicate 2	735.05	986.55	902.56	758.10	864.93	907.99	774.77	866.92	827.34	827.16
	Replicate 3	761.33	989.44	845.09	758.10	864.93	839.28	810.15	798.18	946.35	976.35
	Replicate 4	735.05	989.44	902.56	758.10	864.93	907.99	757.90	843.92	872.20	827.16
	Replicate 5	730.39	986.55	902.56	758.10	776.77	839.28	820.31	816.47	892.91	920.90
	Replicate 6	797.26	986.55	902.56	749.80	864.93	907.99	774.77	873.53	946.35	813.82
	Replicate 7	886.28	986.55	902.56	758.10	864.93	808.88	820.33	892.54	892.91	802.10
	Replicate 8	735.05	986.55	902.56	758.10	864.93	839.28	810.15	767.52	892.91	892.81
	Replicate 9	735.05	986.55	874.35	758.10	864.93	907.99	826.57	879.21	827.34	920.90
	Replicate 10	797.50	989.44	902.56	749.80	864.93	808.88	774.77	816.35	946.35	857.17
Average	Objective	779.92	987.42	892.48	756.37	846.85	859.69	799.00	842.87	893.76	869.55
	S.D.	61.54	1.39	19.18	3.47	38.14	43.03	25.43	41.30	44.21	56.55
Best	Different (%)	-12.00	27.79	9.25	9.61	6.90	15.44	-1.33	-10.91	0.09	-5.50
	Objective	886.28	989.44	902.56	758.10	864.93	907.99	826.57	892.54	946.35	976.35
Best	Different (%)	0.00	28.05	10.48	9.86	9.18	21.93	2.07	-5.66	5.98	6.11

CHULALONGKORN UNIVERSITY

จากการเปรียบเทียบเวลาเฉลี่ยในการหาคำตอบของฮิวริสติก และวิธีแมนตรง พบว่าฮิวริสติกใช้เวลาเฉลี่ยในการหาคำตอบน้อยกว่าวิธีแมนตรง ในทุกตัวอย่าง โดยฮิวริสติกใช้เวลาเฉลี่ยในการหาคำตอบทั้งหมดเท่ากับ 610.07 วินาที (ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 157.64) ในขณะที่วิธีแมนตรงใช้เวลาเฉลี่ยในการหาคำตอบทั้งหมดเท่ากับ 14402.89 วินาที (ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 3.17) ซึ่งเวลาในการหาคำตอบทั้งหมดที่ได้จากการหาคำตอบด้วยฮิวริสติกในหน่วยวินาทีถูกรวบรวมไว้ในตารางที่ 4-13

ตารางที่ 4-13 เวลาในการหาคำตอบด้วยฮิวริสติกในปัญหาที่มี 12 รอบรถ (วินาที)

Solver		Instance									
		1	2	3	4	5	6	7	8	9	10
CPLEX		14400.73	14400.60	14400.71	14400.71	14406.84	14409.33	14404.34	14400.60	14400.76	14404.31
ALNS	Replicate 1	724.35	794.08	728.77	584.64	449.24	741.07	680.23	705.19	417.84	492.23
	Replicate 2	436.43	948.88	502.27	799.95	589.63	533.76	482.74	606.42	446.73	518.31
	Replicate 3	592.91	526.38	461.63	502.17	763.72	647.74	492.98	566.23	561.68	543.08
	Replicate 4	423.49	566.10	460.39	807.16	531.05	531.80	510.30	528.47	420.15	525.02
	Replicate 5	481.91	499.63	672.72	663.68	810.87	663.57	1008.71	432.61	494.35	837.12
	Replicate 6	480.21	561.36	684.93	497.43	674.18	487.31	457.44	1027.73	704.09	702.01
	Replicate 7	914.18	558.94	466.65	523.39	830.15	546.22	691.56	816.38	459.08	512.26
	Replicate 8	474.59	640.88	583.47	752.81	1067.25	545.89	514.77	431.51	469.07	570.58
	Replicate 9	566.06	561.65	532.29	631.68	814.24	572.02	1216.76	616.47	755.53	550.55
	Replicate 10	456.46	563.40	704.42	640.30	870.33	543.05	466.53	540.83	588.08	459.84
	Average	555.06	622.13	579.75	640.32	740.07	581.24	652.20	627.18	531.66	571.10
	S.D.	155.53	140.95	108.89	116.84	182.10	77.67	261.12	182.55	118.84	113.57
Different (%)	-96.15	-95.68	-95.97	-95.55	-94.86	-95.97	-95.47	-95.64	-96.31	-96.04	

ผลการเปรียบเทียบคุณภาพของคำตอบ และเวลาในการหาคำตอบของฮิวริสติกทำให้ได้ข้อสรุปว่า ฮิวริสติกสามารถหาคำตอบที่มีคุณภาพใกล้เคียงกับคำตอบที่ได้จากวิธีแมนตรง ภายใต้การหาคำตอบ 10 ครั้ง ในตัวอย่างทั้งหมด 40 ตัวอย่าง โดยตัวอย่างที่ฮิวริสติกได้คำตอบที่มีคุณภาพต่ำที่สุด คือ ตัวอย่างที่ 8 ของกลุ่มตัวอย่างที่มี 12 รอบรถ ซึ่งค่า Objective ของคำตอบที่ได้จากฮิวริสติก มีค่าต่ำกว่าคำตอบที่ได้จากวิธีแมนตรง เท่ากับ 5.66 เปอร์เซ็นต์

ผู้วิจัยคาดว่า สาเหตุที่ทำให้คุณภาพคำตอบที่ได้จากฮิวริสติกด้อยกว่าวิธีแมนตรง ในบางตัวอย่าง เป็นเพราะ ค่า Cooling rate ของ SA มีค่าน้อยเกินไปในตัวอย่างนั้น ๆ จึงทำให้ฮิวริสติกค้นหาคำตอบได้เพียงภายในเซตที่มีขนาดจำกัด และสิ้นสุดการหาคำตอบในเวลาอันสั้น ดังนั้นโอกาสที่ฮิวริสติกจะพบคำตอบที่มีคุณภาพสูง จึงถูกจำกัดเช่นกัน กล่าวคือ หากต้องการเพิ่มโอกาสพบคำตอบที่มีคุณภาพมากขึ้น จึงควรเพิ่มค่า Cooling rate ของ SA ให้เหมาะสมกับตัวอย่างนั้น ๆ เพื่อให้การค้นหาคำตอบละเอียดมากขึ้น ในขณะที่เวลาการหาคำตอบไม่มากจนเกินไป

บทที่ 5

สรุปผลการดำเนินงานวิจัย

งานวิจัยนี้ ผู้วิจัยได้สร้างแบบจำลองทางคณิตศาสตร์ และวิธีการฮิวริสติกสำหรับปัญหา SDPD แบบมีจำนวนรถจำกัด จากการอ้างอิงปัญหา DDWP แบบ Deterministic และปัญหา PDP โดยมีวัตถุประสงค์เพื่อเพิ่มกำไรสุทธิจากการดำเนินงาน และเป็นขั้นตอนสำคัญที่จะนำไปสู่การพัฒนาวิธีแก้ปัญหา SDPD แบบที่มีความไม่แน่นอนของคำสั่งของลูกค้า หรือมีข้อจำกัดด้านทรัพยากรอื่น ๆ ในงานวิจัยในอนาคต

5.1 ผลการวิจัย

เพื่อความง่ายในการจำลองปัญหา SDPD ผู้วิจัยจึงกำหนดระยะเวลาทำการเท่ากับ 10 ชั่วโมง ให้ใกล้เคียงกับระยะเวลาทำการมาตรฐานของบริษัทขนส่งโดยทั่วไป จากนั้นจึงสุ่มเวลาการปรากฏของแต่ละคำสั่งให้มีค่าอยู่ระหว่างรอบรถแรกกับรอบรถที่อยู่กึ่งกลางของระยะเวลาทำการด้วยการกระจายตัวแบบสมมาตร และสุ่มตำแหน่งของคำสั่งของลูกค้าให้อยู่ห่างจากท่ารถไม่เกิน 1.25 ชั่วโมง ด้วยการกระจายตัวแบบสมมาตร เพื่อให้ทุกคำสั่งมีความเป็นไปได้ที่จะได้รับบริการภายในวันทำการเดียวกัน

เนื่องจากสมมติฐานของปัญหากำหนดให้มีรถใช้งานเพียงคันเดียว ดังนั้นระดับการให้บริการจึงถูกจำกัดด้วยระยะเวลาทำการ ผู้วิจัยจึงทดลองหาขนาดที่เหมาะสมกับสมมติฐาน โดยการสุ่มตัวอย่างที่มี 6 รอบรถ และมีจำนวนคำสั่งของลูกค้าต่อวันเท่ากับ 10, 12, 14, 16, 18, 20, 22, และ 24 คำสั่ง เพื่อหาคำตอบที่ดีที่สุดด้วยวิธีแมนตรง ภายในระยะเวลาจำกัด 4 ชั่วโมง ซึ่งผลการทดลองทำให้ได้ข้อสรุปว่า ปัญหาที่มีขนาดเหมาะสมสำหรับการตั้งค่างานวิจัย ประกอบด้วยคำสั่งของลูกค้าจำนวนไม่เกิน 18 คำสั่งของลูกค้า ซึ่งมีระดับการให้บริการเฉลี่ยประมาณ 60 เปอร์เซ็นต์

นอกจากนี้ เนื่องจากปัญหา SDPD มีความซับซ้อนรูปแบบ NP-hard ดังนั้นผู้วิจัยจึงทดลองหาขีดจำกัดความสามารถของการหาคำตอบที่ดีที่สุดด้วยวิธีแมนตรง ด้วยการเพิ่มความซับซ้อนของปัญหา โดยเพิ่มจำนวนรอบรถต่อวันจาก 6 รอบ เป็น 8, 10, และ 12 รอบ ซึ่งผลการทดลองแสดงให้เห็นว่า การเพิ่มจำนวนรอบรถต่อวันเพียง 2 รอบ สามารถทำให้ความสามารถของการหาคำตอบด้วยวิธีแมนตรง ลดลง สังเกตได้จาก %GAP เฉลี่ยของกลุ่มตัวอย่างที่มี 6, 8, 10, และ 12 รอบรถ ซึ่งมีค่าเท่ากับ 3.79, 10.73, 18.76, และ 27.99 เปอร์เซ็นต์ ตามลำดับ ด้วยเหตุนี้ผู้วิจัยจึงพัฒนาฮิวริสติกสำหรับหาคำตอบของปัญหา SDPD โดยคาดหวังคำตอบที่มีคุณภาพเทียบเท่า หรือใกล้เคียงคำตอบที่ได้จากวิธีแมนตรง และใช้เวลาการหาคำตอบน้อยกว่าวิธีแมนตรง

ฮิวริสติกรูปแบบ ALNS ที่ถูกพัฒนาขึ้นสำหรับหาคำตอบของปัญหา SDPD ถูกทดสอบประสิทธิภาพในการหาคำตอบของฮิวริสติก ทั้งในด้านคุณภาพของคำตอบ และเวลาในการหาคำตอบ ด้วยการนำตัวอย่างจากการทดลองหาซิดจำกัดความสามารถของการหาคำตอบที่ดีที่สุดด้วยวิธีแมนตรง มาหาคำตอบด้วยฮิวริสติกทั้งหมด 10 ครั้ง ต่อ 1 ตัวอย่าง และนำคำตอบที่ดีที่สุดที่ได้จากฮิวริสติกมาเปรียบเทียบกับคำตอบที่ได้จากวิธีแมนตรง

ผลการทดสอบประสิทธิภาพของฮิวริสติกในด้านคุณภาพของคำตอบ แสดงให้เห็นว่า จากตัวอย่างทั้งหมด 40 ตัวอย่าง ค่า Objective ของคำตอบที่ดีที่สุดจากฮิวริสติกมีค่าเท่ากับคำตอบที่ได้จากวิธีแมนตรง ทั้งหมด 15 ตัวอย่าง, มากกว่าคำตอบที่ได้จากวิธีแมนตรง ทั้งหมด 21 ตัวอย่าง, และน้อยกว่าคำตอบที่ได้จากวิธีแมนตรง ทั้งหมด 4 ตัวอย่าง โดยคำตอบที่มีคุณภาพต่ำที่สุดในตัวอย่างที่มี 6, 8, 10, และ 12 รอบรถ มีค่า Objective ต่ำกว่าคำตอบที่ได้จากวิธีแมนตรง เพียง 1.27, 1.20, 1.62, และ 5.66 เปอร์เซ็นต์ ตามลำดับ

ผลการทดสอบประสิทธิภาพของฮิวริสติกในด้านเวลาในการหาคำตอบ แสดงให้เห็นว่า ฮิวริสติกใช้เวลาเฉลี่ยในการหาคำตอบของตัวอย่างที่มี 6, 8, 10, และ 12 รอบรถ เท่ากับ 420.29, 474.49, 537.61, 610.07 วินาที ตามลำดับ ในขณะที่ OPL ใช้เวลาหาคำตอบเฉลี่ยเท่ากับ 6999, 10770, 13748, 14403 วินาที ตามลำดับ ทั้งนี้จึงสรุปได้ว่า ฮิวริสติกสามารถให้คำตอบที่มีคุณภาพใกล้เคียงวิธีแมนตรง โดยใช้เวลาในการหาคำตอบน้อยกว่า

5.2 แนวทางสำหรับพัฒนาวิธีการแก้ปัญหา SDPD ที่มีความไม่แน่นอนของคำสั่งของลูกค้า

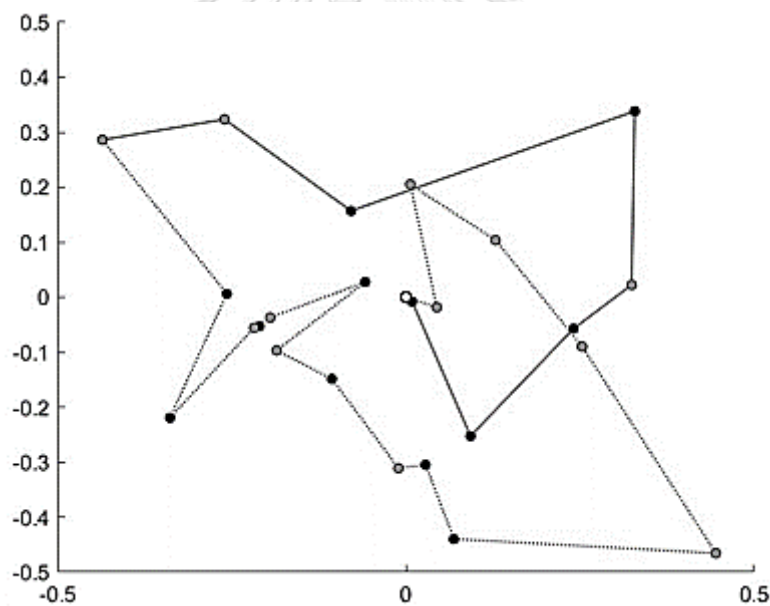
ปัญหา SDPD ที่มีความไม่แน่นอนของคำสั่งของลูกค้า (Stochastic SDPD) จัดเป็นปัญหา Dynamic VRP รูปแบบหนึ่ง ที่ต้องหาคำตอบอย่างพลวัต (Real-time solving) เพื่อปรับปรุงเส้นทางเดินรถด้วยคำสั่งลูกค้าที่ปรากฏขึ้นใหม่ภายในวันทำการ โดยการแก้ปัญหาดังกล่าวอาจทำได้โดย การนำคำตอบของการแก้ปัญหา Deterministic SDPD มาสร้างคำตอบเบื้องต้น (*a priori* solution) ของปัญหา Stochastic SDPD จากนั้นจึงออกแบบนโยบายพลวัตสำหรับแก้ปัญหา Stochastic SDPD ด้วยคำตอบเบื้องต้นที่สร้างขึ้น เช่นเดียวกับที่ Klapp et al., (2018) ได้ทำการออกแบบนโยบายพลวัตสำหรับแก้ปัญหา SDD อย่างพลวัต หรือการแยกปัญหา Stochastic SDPD ออกเป็นอนุกรมของ Static SDPD จากนั้นจึงหาคำตอบของแต่ละปัญหาย่อยด้วยวิธีฮิวริสติก (Euchi et al., 2015)

อย่างไรก็ดี เนื่องจากในแต่ละรอบของการแก้ปัญหา Stochastic SDPD อย่างพลวัต จำเป็นต้องพิจารณาข้อมูลคำสั่งของลูกค้าที่ปรากฏขึ้นแล้ว (Open orders) และคำสั่งลูกค้าที่ต้องให้บริการเนื่องจากรับพัสดุมานแล้ว (Close orders) ดังนั้นจึงต้องใช้เซตของคำสั่งของลูกค้าที่ปรากฏ

ขึ้นแล้ว และเซตของเส้นทางที่ไม่สามารถแก้ไขได้ เป็นข้อมูลนำเข้าในการแก้ปัญหาแต่ละรอบ โดยสามารถดัดแปลงแบบจำลองทางคณิตศาสตร์สำหรับปัญหา Deterministic SDPD ในงานวิจัยนี้ ด้วยการเพิ่มข้อจำกัดสำหรับตรงเส้นทางที่รถได้เดินทางผ่านไปแล้ว ดังสมการที่ (5.1) เมื่อ A' คือ เซตของเส้นทางที่รถเดินทางผ่านไปแล้ว และ w_a คือ รอบรถที่ใช้เส้นทาง a

$$x_a^{w_a} = 1 \quad \forall a \in A' \quad (5.1)$$

ทั้งนี้แผนภาพการปล่อยรถที่เป็นคำตอบหนึ่งของการแก้ปัญหา Stochastic SDPD ด้วยวิธีพลวัต ถูกแสดงไว้ในรูปที่ 5-1 เมื่อเส้นทึบเป็นเส้นทางที่รถเดินทางผ่านไปแล้ว ที่ไม่สามารถเปลี่ยนแปลงได้ และเส้นประเป็นเส้นทางที่สามารถเปลี่ยนแปลงได้ในรอบการหาคำตอบถัดไป หากรถยังไม่เดินทางบนเส้นทางนั้น ๆ



รูปที่ 5-1 ตัวอย่างคำตอบของปัญหา Stochastic SDPD

5.3 ข้อเสนอแนะ

เนื่องจากในงานวิจัยนี้ผู้วิจัยได้กำหนดสมมติฐานให้มีรถสำหรับรับ-ส่งพัสดุเพียงคันเดียว ดังนั้นเพื่อให้แบบจำลองมีความใกล้เคียงกับปัญหาจริงมากขึ้น จึงควรกำหนดให้มีรถได้มากกว่าหนึ่งคัน โดยการดัดแปลงตัวแปรตัดสินใจ และสมการข้อกำหนดให้สอดคล้องกับรูปแบบปัญหาที่เป็น Multi-vehicle Pickup and Delivery Problem และเพิ่มพจน์ของค่าใช้จ่ายจากการใช้รถในสมการวัตถุประสงค์ตามสมการที่ (5.2) เมื่อ V คือ ค่าใช้จ่ายจากการใช้งานรถหนึ่งคัน

$$C^* = \max_{(x,y,v,z)} \sum_{r \in R} \sum_{w \in W_{p_r}}^{\tau_r} P_r y_{p_r}^w - \gamma \sum_{w \in W} \sum_{a \in A_w} t_a x_a^w + V \left(\sum_{w \in W} \sum_{j \in N_w} x_{\emptyset_j}^w \right) \quad (5.2)$$

นอกจากนี้ ในงานวิจัยนี้ได้ศึกษาเพียงปัญหา Deterministic SDPD ดังนั้นการวิจัยปัญหา Stochastic SDPD ที่มีความไม่แน่นอนของคำสั่งของลูกค้า จึงมีความสำคัญสำหรับการศึกษาปัญหา SDPD ในอนาคต โดยอาจประยุกต์แนวทางจากหัวข้อ 5.2 สำหรับพัฒนาวิธีการแก้ปัญหา Stochastic SDPD แบบพลวัต



บรรณานุกรม

- AFSHAR-NADJAFI, B. & AFSHAR-NADJAFI, A. 2017. A constructive heuristic for time-dependent multi-depot vehicle routing problem with time-windows and heterogeneous fleet. *Journal of King Saud University - Engineering Sciences*, 29, 29-34.
- AVCI, M. G. & AVCI, M. 2019. An adaptive large neighborhood search approach for multiple traveling repairman problem with profits. *Computers & Operations Research*, 111, 367-385.
- CACCHIANI, V., CONTRERAS-BOLTON, C. & TOTH, P. 2020. Models and algorithms for the Traveling Salesman Problem with Time-dependent Service times. *European Journal of Operational Research*, 283, 825-843.
- CLARKE, G. & WRIGHT, J. W. 2015. Scheduling of vehicles from a central depot to a number of delivery points. *Operation Research*, 12, 568-581.
- EUCHI, J., YASSINE, A. & CHABCHOUB, H. 2015. The dynamic vehicle routing problem: Solution with hybrid metaheuristic approach. *Swarm and Evolutionary Computation*, 21, 41-53.
- GHILAS, V., DEMIR, E. & WOENSEL, T. V. 2016. An adaptive large neighborhood search heuristic for the Pickup and Delivery Problem with Time Windows and Scheduled Lines. *Computers & Operations Research*, 72, 12-30.
- GU, W., CATTARUZZA, D., OGIER, M. & SEMET, F. 2019. Adaptive large neighborhood search for the commodity constrained split delivery VRP. *Computers & Operations Research*, 112.
- HAJJAM, A., CRÉPUT, J. C. & KOUKAM, A. 2013. *From the TSP to the Dynamic VRP: An Application of Neural Networks in Population Based Metaheuristic*. In: Alba E., Nakib A., Siarry P. (eds) *Metaheuristics for Dynamic Optimization. Studies in Computational Intelligence*, Springer, Berlin, Heidenberg.
- HORNSTRA, R. P., SILVA, A., ROODBERGEN, K. J. & COELHO, L. C. 2020. The vehicle routing problem with simultaneous pickup and delivery and handling costs. *Computers & Operations Research*, 115.

- HOUGARDY, S., ZAISER, F. & ZHONG, X. 2020. The approximation ratio of the 2-Opt Heuristic for the metric Traveling Salesman Problem. *Operations Research Letters*, 48, 401-404.
- KLAPP, M. A., ERERA, A. L. & TORIELLO, A. 2018. The Dynamic Dispatch Waves Problem for same-day delivery. *European Journal of Operational Research*, 271, 519-534.
- LEWIS, H. F. & SEXTON, T. R. 2007. On the tour partitioning heuristic for the unit demand capacitated vehicle routing problem. *Operations Research Letters*, 35, 374-378.
- LI, Y., CHEN, H. & PRINS, C. 2016. Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests. *European Journal of Operational Research*, 252, 27-38.
- MAHI, M., BAYKAN, O. K. & KODAZ, H. 2015. A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem. *Applied Soft Computing*, 30, 484-490.
- MARTINS, L. C., BAYLISS, C., JUAN, A. A., PANADERO, J. & MARMOL, M. 2020. A Savings-Based Heuristic for Solving the Omnichannel Vehicle Routing Problem with Pickup and Delivery. *Transportation Research Procedia*, 47, 83-90.
- PUREZA, V., MORABITO, R. & REIMANN, M. 2012. Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the VRPTW. *European Journal of Operational Research*, 218, 636-647.
- SACRAMENTO, D., PISINGER, D. & ROPKE, S. 2019. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies*, 102, 289-315.
- SANTOS, F. A., MATEUS, G. R. & CUNHA, A. S. 2013. The Pickup and Delivery Problem with Cross-Docking. *Computers & Operations Research*, 40, 1085-1093.
- SHERIDAN, P. K., GLUCK, E., GUAN, Q., PICKLES, T., BALCIOĞLU, B. & BENHABIB, B. 2013. The dynamic nearest neighbor policy for the multi-vehicle pick-up and delivery problem. *Transportation Research Part A: Policy and Practice*, 49, 178-194.
- SONG, M., LI, J., HAN, Y., HAN, Y., LIU, L. & SUN, Q. 2020. Metaheuristics for solving the vehicle routing problem with the time windows and energy consumption in cold chain logistics. *Applied Soft Computing*, 95.

STATISTA. Available: <http://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/> [Accessed May 17 2019].

SUN, P., VEELNTURF, L. P., HEWITT, M. & WOENSEL, T. V. 2020. Adaptive large neighborhood search for the time-dependent profitable pickup and delivery problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, 138.





จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ประวัติผู้เขียน

ชื่อ-สกุล	วิษณะ ศรีตระกูล
วัน เดือน ปี เกิด	28 เมษายน 2538
วุฒิการศึกษา	สำเร็จการศึกษาหลักสูตรปริญญาบัณฑิต สาขาเทคโนโลยีทางอาหาร แขนง วิชาเทคโนโลยีทางอาหาร จุฬาลงกรณ์มหาวิทยาลัย ในปีพุทธศักราช 2561 และเข้ารับการศึกษาคือต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขา วิศวกรรมอุตสาหกรรม ในปีพุทธศักราช 2561
ที่อยู่ปัจจุบัน	895/152 ซอยปทุมวันวิที48 ถนนสุขุมวิท101 แขวงบางจาก เขตพระโขนง จังหวัดกรุงเทพมหานคร 10260



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY